

Administration guide

Contents

Articles

Administration Guide	1
Administration access	2
Configuration view	3
Mime types	3
Statistics	5
Scripting - OpenKM 6.2	7
Property Groups administration	11
Property Groups	13
Logged users	16
Message queue	16
User administration	19
Profiles	28
Database query	30
Reports	30
Activity log	34
Workflow	35
Automation	37
Enable automation	44
Extend automation	49
Extend automation 6.4	49
Extend automation 6.0	52
Crontab	54
OMR templates	58
OCR templates	67
Extend OCR field parsers	72
Thesaurus	74
Folder style admin	75
Generate thesaurus	81
Language	82
Repository export	84
Utilities admin	84
Utilites admin check mail	85
Application configuration	86

References

Article Sources and Contributors	99
Image Sources, Licenses and Contributors	100

Administration Guide

OpenKM is a tool to manage documents generated within a company, to improve access within the whole corporate environment and to endeavour to keep in check the unnecessary duplication of information.

OpenKM administration provides the solution to:

- Control the growing volume of information to be managed
- Manage users
- View actual OpenKM.cfg configuration file properties loaded by OpenKM at starting up.
- Scripting
- Low level access to repository
- Metadata
- Queries and reporting tools
- Auditing users (track log)
- Managing the workflow
- Importing and exporting data from your server
- Managing thesaurus

Through the OpenKM administration system, relevant and useful information can be accessed to help administrator in controlling the OpenKM repository.

- Administration access
 - Configuration view
 - Mime types
 - Statistics
 - Scripting
 - OpenKM 6.2
 - OpenKM 5.1
 - Property Groups administration
 - Logged users
 - Message queue
 - User administration
 - Profiles
 - Database query
 - Reports
 - Activity log
 - Workflow
 - Automation
 - Crontab
 - OMR templates
 - OCR templates
 - Folder style
 - Generate thesaurus
 - Language
 - Repository import
 - Repository export
 - Utilities
-

OpenKM 5.1 and older

- Administration search
 - XPath queries
 - SQL queries
- Repository view

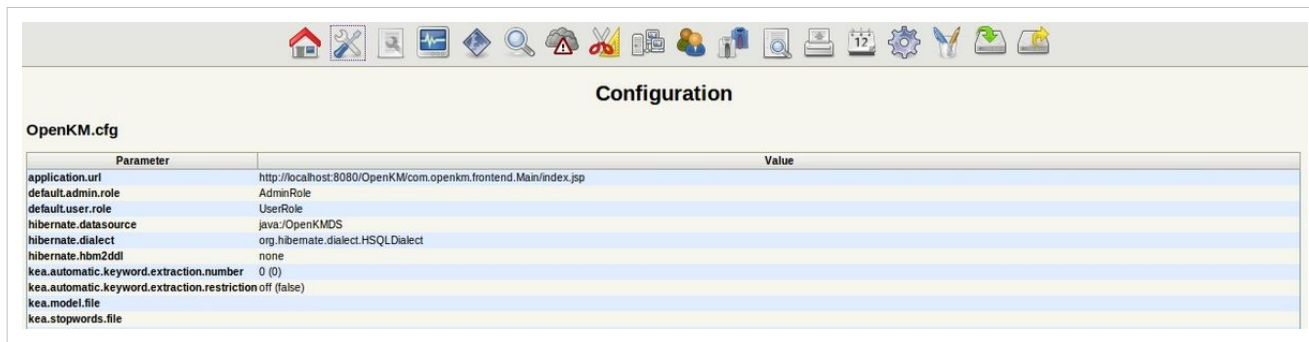
Administration access

Only users that have the role **AdminRole** can access administration. A new tab **Administration** will appear for these users at the top right to access administration utilities.



Configuration view

In configuration view, an administrator can see the parameters that have been loaded during OpenKM start up. This has special interest when something has been changed in OpenKM.cfg, to see if really the parameters have been loaded correctly during the start up sequence.



The screenshot shows the 'Configuration' window with a toolbar at the top. Below the title 'Configuration', there is a sub-header 'OpenKM.cfg'. A table lists various parameters and their values.

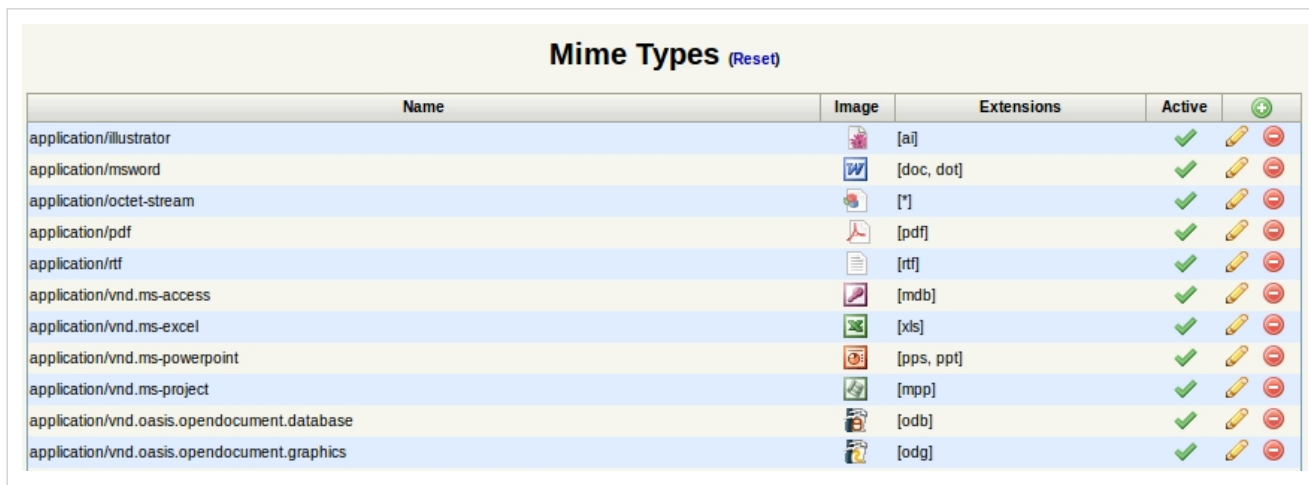
Parameter	Value
application.url	http://localhost:8080/OpenKM/com.openkm.frontend.Main/index.jsp
default.admin.role	AdminRole
default.user.role	UserRole
hibernate.datasource	java:/OpenKMDS
hibernate.dialect	org.hibernate.dialect.HSQLDialect
hibernate.hbm2ddl	none
kea.automatic.keyword.extraction.number	0 (0)
kea.automatic.keyword.extraction.restriction	off (false)
kea.model.file	
kea.stopwords.file	

Mime types

OpenKM supported mime type files can be added, removed or edited.

Detail of the functions according to their icons:

- ➕ → Create new mime type
- ✓ → Mime type enabled
- ✗ → Mime type disabled
- ✎ → Edit mime type
- ⊖ → Delete mime type

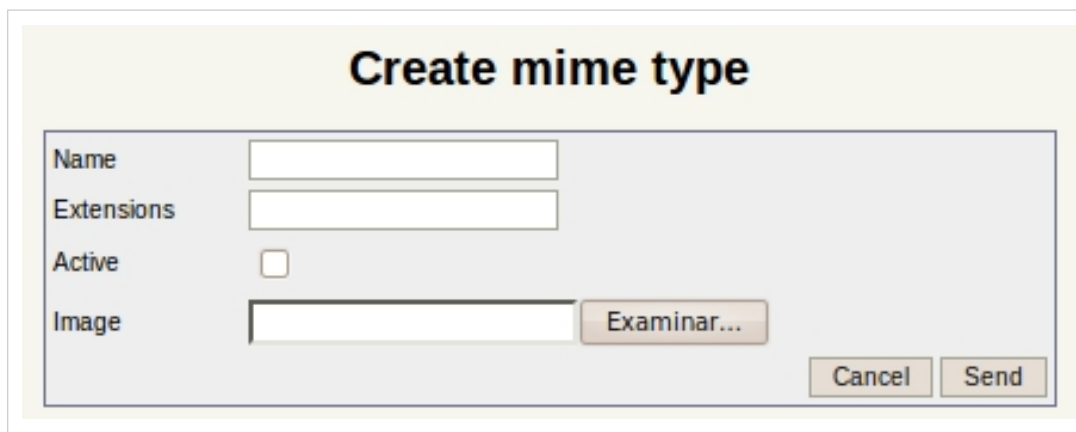


The screenshot shows the 'Mime Types (Reset)' window. It contains a table with columns for Name, Image, Extensions, Active, and action icons (plus, edit, delete).

Name	Image	Extensions	Active	
application/illustrator		[ai]	✓	✎ ⊖
application/msword		[doc, dot]	✓	✎ ⊖
application/octet-stream		[*]	✓	✎ ⊖
application/pdf		[pdf]	✓	✎ ⊖
application/rtf		[rtf]	✓	✎ ⊖
application/vnd.ms-access		[mdb]	✓	✎ ⊖
application/vnd.ms-excel		[xls]	✓	✎ ⊖
application/vnd.ms-powerpoint		[pps, ppt]	✓	✎ ⊖
application/vnd.ms-project		[mpp]	✓	✎ ⊖
application/vnd.oasis.opendocument.database		[odb]	✓	✎ ⊖
application/vnd.oasis.opendocument.graphics		[odg]	✓	✎ ⊖

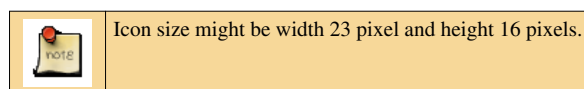
Create a new mime type

To create a mime type, click on  the create new mime type icon:




The 'Create mime type' dialog box has a title bar with the text 'Create mime type'. Inside, there are four labeled fields: 'Name' with an empty text box, 'Extensions' with an empty text box, 'Active' with an unchecked checkbox, and 'Image' with an empty text box. To the right of the 'Image' field is a button labeled 'Examinar...'. At the bottom right of the dialog are two buttons: 'Cancel' and 'Send'.

After creating the new mime type, you'll return to view all mime type list (now with the mime type you've created).



Delete mime type


To delete a mime type, click on  the delete mime type icon:

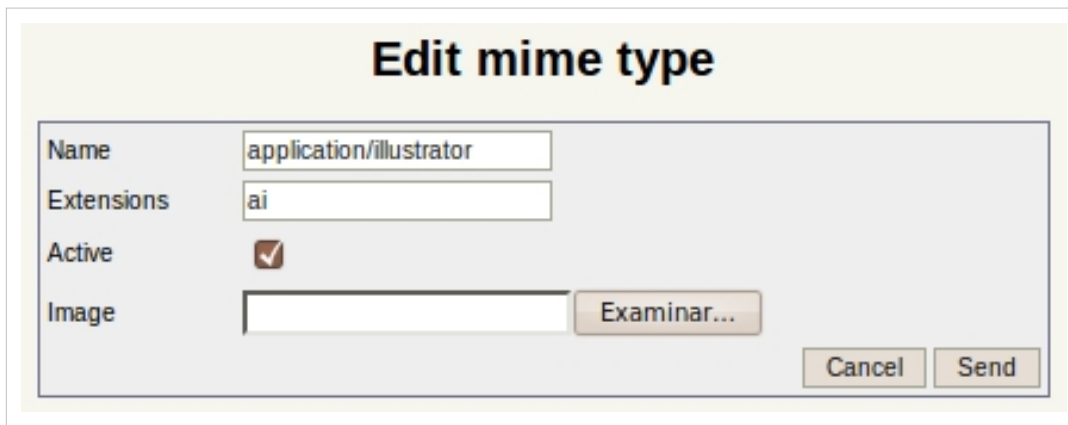


The 'Delete mime type' dialog box has a title bar with the text 'Delete mime type'. Inside, there are four labeled fields: 'Name' with the text 'application/illustrator', 'Extensions' with the text 'ai', 'Active' with a checked checkbox, and 'Image' with an empty text box. To the right of the 'Image' field is a button labeled 'Examinar...'. At the bottom right of the dialog are two buttons: 'Cancel' and 'Send'.

After deleting a mime type, you'll return to view all mime type list.

Edit mime type

To edit a mime type, click on  the edit mime type icon:



Edit mime type

Name	<input type="text" value="application/illustrator"/>
Extensions	<input type="text" value="ai"/>
Active	<input checked="" type="checkbox"/>
Image	<input type="text"/> <input type="button" value="Examiner..."/>

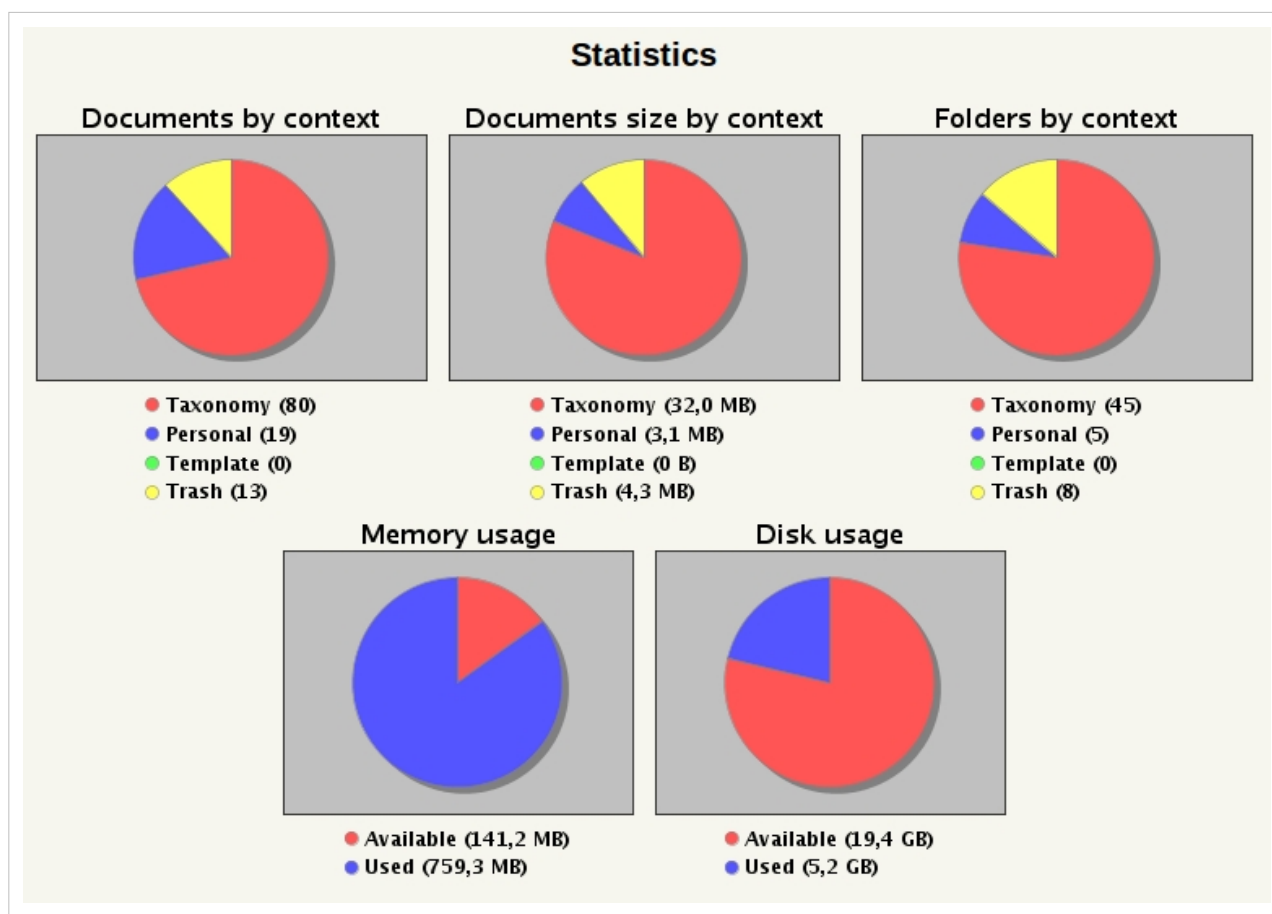
After updating a mime type, you'll return to view all mime type list.

Statistics

In statistics view, administrators can view information about repository size and it's distribution.

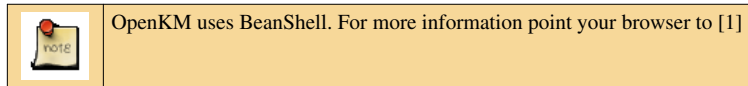
Of special interest:

- Document size by contents (to understand how your repository is growing and making hard disk growth estimations)
- Memory usage (to understand OpenKM memory usage).



Scripting - OpenKM 6.2

Scripting was an advanced feature introduced in OpenKM 5.0 that enables administrators to execute some BeanShell scripts in folders, fired on every notified event (for example uploading documents). This feature has been overseed by Automation, which help you to do the same things with a few mouse clicks. For compatibility reasons there is also an Automation task where you can put a BeanShell code. This replaces the old way of enabling scripting.



BeanShell is a small, free, embeddable Java source interpreter with object scripting language features, written in Java. BeanShell dynamically executes standard Java syntax and extends it with common scripting conveniences such as loose types, commands, and method closures like those in Perl and JavaScript.

You can also execute a custom script when OpenKM starts or stops. To do so, just create a **start.bsh** (and / or **stop.bsh**) file in \$TOMCAT_HOME. For example, we use this feature to create a complete environment (create custom users, register property groups, register workflow) each time the OpenKM demo is restarted.

Examples

Purge all users trash

```
import com.openkm.api.*;
import com.openkm.core.*;
import com.openkm.bean.*;
import com.openkm.module.db.stuff.*;

String token = DbSessionManager.getInstance().getSystemToken();

for (Folder trash : OKMFolder.getInstance().getChilds(token,
"/okm:trash")) {
    print("Trash: " + trash.getPath()+"<br/>");

    for (Folder fld : OKMFolder.getInstance().getChilds(token,
trash.getPath())) {
        print("About to delete folder: " + fld.getPath() + "<br/>");
        OKMFolder.getInstance().purge(token, fld.getPath());
    }

    for (Document doc : OKMDocument.getInstance().getChilds(token,
trash.getPath())) {
        print("About to delete document: " + doc.getPath() + "<br/>");
        OKMDocument.getInstance().purge(token, doc.getPath());
    }

    for (Mail mail : OKMMail.getInstance().getChilds(token,
trash.getPath())) {
        print("About to delete mail: " + mail.getPath() + "<br/>");
        OKMMail.getInstance().purge(token, mail.getPath());
    }
}
```

```

    }
}

```

Update repository stats

```

import com.openkm.core.*;

new RepositoryInfo().run();

```

Import reports from a folder

Import reports from \$TOMCAT_HOME/reports and also enable them in the default profile.

```

import java.io.*;
import java.sql.*;
import org.hibernate.*;
import com.openkm.core.*;
import com.openkm.util.*;
import com.openkm.dao.*;

Session hbmSession = HibernateUtil.getSessionFactory().openSession();
Connection con = hbmSession.connection();
Statement st = con.createStatement();
File reports = new File(Config.HOME_DIR + "/reports");

try {
    if (reports.isDirectory()) {
        for (File rep : reports.listFiles()) {
            int id = ReportDAO.createFromFile(rep,
FileUtils.getFileName(rep.getName()), true);
            String sql = "insert into OKM_PROFILE_MSC_REPORT (PRP_ID,
PRP_REPORT) values (1, " + id + ")";
            LegacyDAO.execute(con, sql);
        }
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    LegacyDAO.close(con);
    HibernateUtil.close(hbmSession);
}

```

Show jBPM mail configuration

```

import org.jbpm.*;

print("Templates: " +
JbpmConfiguration.Configs.getString("resource.mail.templates") + "<br/>");
print("From: " +
JbpmConfiguration.Configs.getString("jbpm.mail.from.address") + "<br/>");

```

```
print("Host: " +  
JbpmConfiguration.Configs.getString("jbpm.mail.smtp.host") + "<br/>");
```

Reset user document size

```
import com.openkm.dao.bean.cache.*;  
import com.openkm.cache.*;  
  
UserItems ui = UserItemsManager.get("okmAdmin");  
ui.setSize(0);
```

Refresh user items

```
import com.openkm.cache.*;  
  
UserItemsManager.refreshDbUserItems();
```

Create some folders and set property group

```
import com.openkm.api.*;  
  
for (int i=0; i < 10; i++) {  
    String path = "/okm:root/fld_" + i;  
    OKMFolder.getInstance().createSimple(null, path);  
    OKMPropertyGroup.getInstance().addGroup(null, path, "okg:technology");  
}
```

Show number of documents, folders and size from a given path

```
import com.openkm.bean.*;  
import com.openkm.util.*;  
import com.openkm.api.*;  
  
ContentInfo ci = OKMFolder.getInstance().getContentInfo(null,  
"/okm:root/path/to/folder");  
print("Folders: " + ci.getFolders() + "<br/>");  
print("Documents: " + ci.getDocuments() + "<br/>");  
print("Size: " + FormatUtil.formatSize(ci.getSize()) + "<br/>");
```

Get path by UUID

```
import com.openkm.api.OKMRepository;  
  
String path = OKMRepository.getInstance().getNodePath(null,  
"b99f7973-4b90-457a-a2d0-cf2090ba995d");  
print(path);
```

Compact documents with size=0

Find all documents with size=0 and compact history of versions to latest

```
import java.util.*;  
import com.openkm.api.OKMDocument;
```

```

import com.openkm.api.OKMRepository;
import com.openkm.dao.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.Query;
import com.openkm.dao.bean.NodeDocumentVersion;

String qs = "from NodeDocumentVersion ndv where ndv.size=0";
Session session = HibernateUtil.getSessionFactory().openSession();
Transaction tx = session.beginTransaction();
Query q = session.createQuery(qs);
List docVersionList = q.list();
HibernateUtil.commit(tx);

print("Number of nodes"+docVersionList.size()+"<br/>");

for (NodeDocumentVersion ndv : docVersionList) {
    String path =
OKMRepository.getInstance().getNodePath(null, ndv.getParent());
    OKMDocument.getInstance().purgeVersionHistory(null, path);
}

HibernateUtil.close(session);
print("done");

```

Recursive repository traversal

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.openkm.api.*;
import com.openkm.bean.*;

Logger log = LoggerFactory.getLogger("com.openkm.scripting");
int MAX_DEPTH = Integer.MAX_VALUE;

void nodeTask(String path, int depth) throws Exception {
    for (Document doc : OKMDocument.getInstance().getChildren(null,
path)) {
        log.info("Document: {}", doc.getPath());
    }

    for (Folder fld : OKMFolder.getInstance().getChildren(null, path))
    {
        log.info("Folder: {}", fld.getPath());

        if (depth < MAX_DEPTH) {
            nodeTask(fld.getPath(), depth + 1);
        }
    }
}

```

```

    }
}

log.info("***** Process BEGIN *****");
nodeTask("/okm:root", 0);
log.info("***** Process END *****");

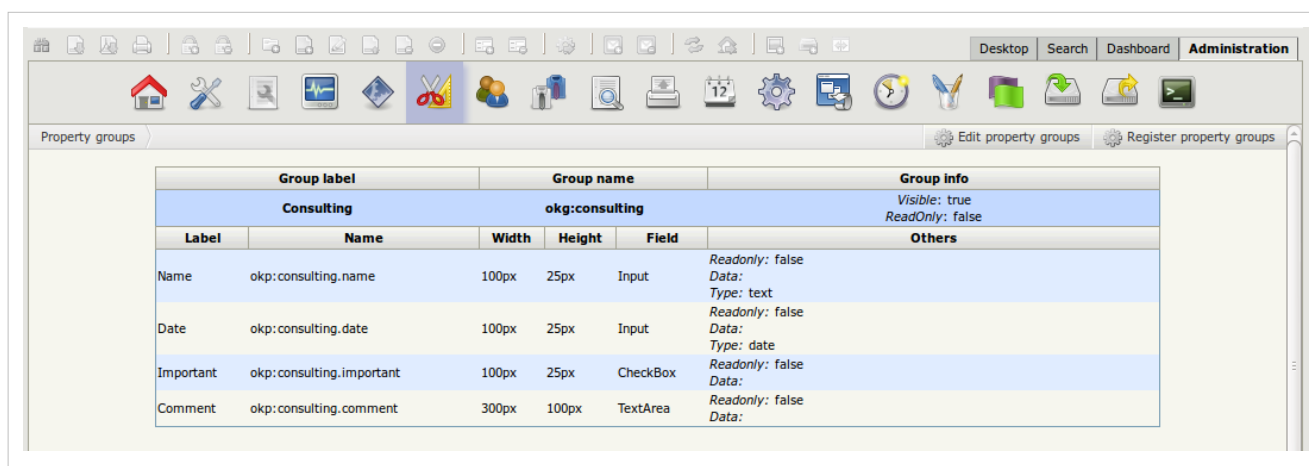
```

References

[1] <http://www.beanshell.org/intro.html>.

Property Groups administration

In Property Groups view (this is the way to add metadata to documents) is shown the registered group properties. In this view, new metadata properties can be registered.

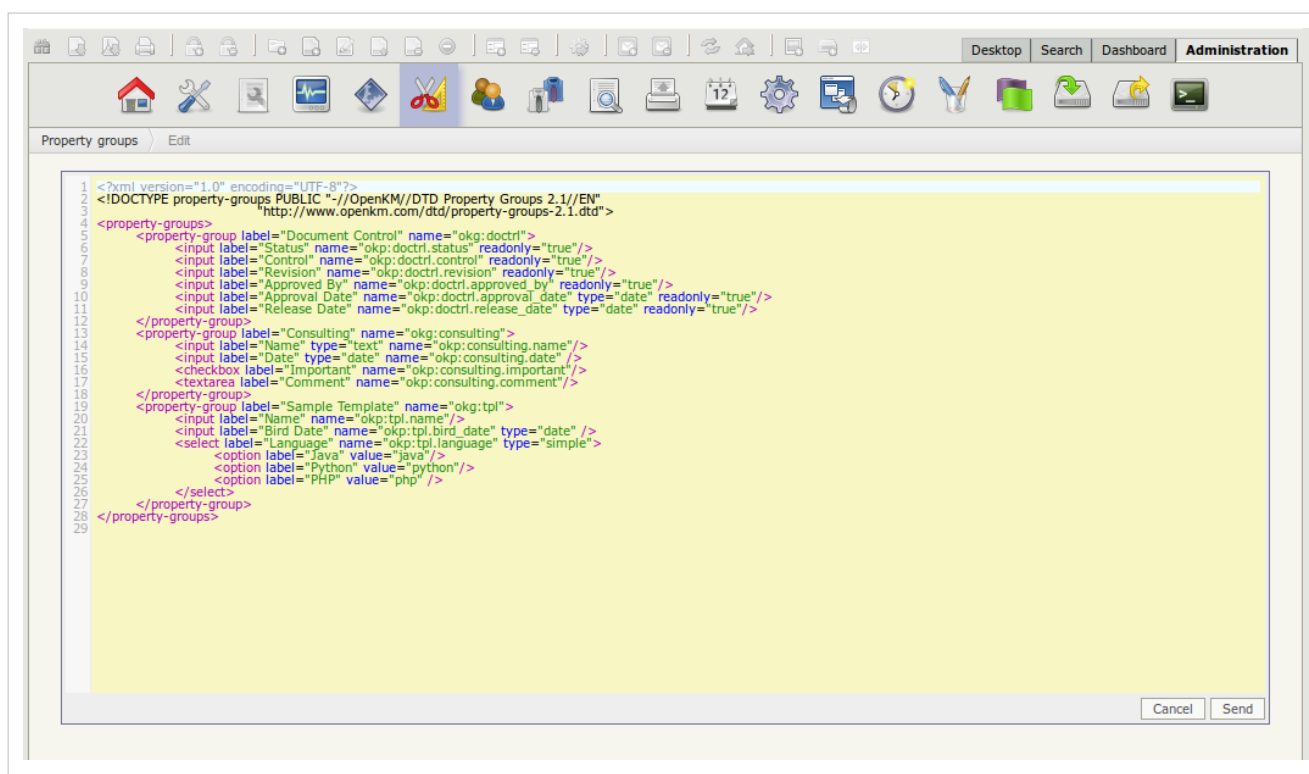


The screenshot displays the 'Property groups' view in OpenKM. The interface includes a toolbar with various icons and a sidebar with tabs like 'Desktop', 'Search', 'Dashboard', and 'Administration'. The main area shows a table of registered properties for the 'Consulting' group.

Group label		Group name			Group info	
Consulting		okg:consulting			Visible: true ReadOnly: false	
Label	Name	Width	Height	Field	Others	
Name	okp:consulting.name	100px	25px	Input	ReadOnly: false Data: text Type: text	
Date	okp:consulting.date	100px	25px	Input	ReadOnly: false Data: date Type: date	
Important	okp:consulting.important	100px	25px	CheckBox	ReadOnly: false Data:	
Comment	okp:consulting.comment	300px	100px	TextArea	ReadOnly: false Data:	

Managing Property Groups

Since OpenKM 6.2 you can edit the Property Group XML definition directly from Administration:



Once you have modified the Property Definition you can register it clicking in the **Register property group** button.



For more information on how to create property groups, take a look at Property Groups in the Installation Guide.

Registering Property Group in OpenKM 5.1

To register a new property group, simply set the file path in the OpenKM server.

Register property group

Property Group definition path

Send

Property Groups

This is an easy and powerful way for building custom properties. You can use this feature to add information to documents. For example, if you store the invoices in a folder you can add a property group to set the client name and the quantity. You will see this important data without opening the documents. Also you can search by these custom properties later.



Refer to Property Groups definition where you can see the new way to describe Property Groups.

By default sample Property Groups are bundle with your OpenKM installation. If you don't see these Property Groups, click in the **Register property groups** button to register them.

Group label	Group name	Group info			
Consulting	okg:consulting	Visible: true ReadOnly: false			
Label	Name	Width	Height	Field	Others
Name	okp:consulting.name	100px	25px	Input	ReadOnly: false Data: text Type: text
Date	okp:consulting.date	100px	25px	Input	ReadOnly: false Data: date Type: date
Important	okp:consulting.important	100px	25px	CheckBox	ReadOnly: false Data: boolean
Comment	okp:consulting.comment	300px	100px	TextArea	ReadOnly: false Data: text



You can also add, remove or modify these Property Groups editing the **PropertyGroups.xml** definition file with your preferred editor or directly from OpenKM Administration if you click in the **Edit property groups** button. Once edited you need to register them again.

```

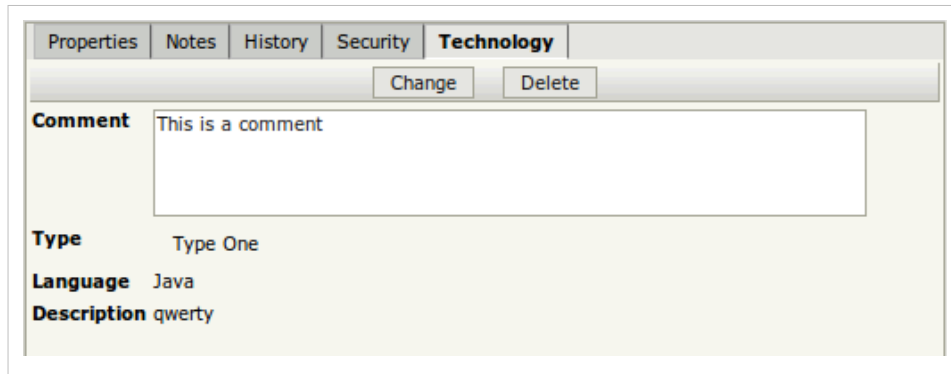
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE property-groups PUBLIC "-//OpenKM//DTD Property Groups 2.1/EN"
3 "http://www.openkm.com/dtd/property-groups-2.1.dtd">
4 <property-groups>
5   <property-group label="Document Control" name="okg:doctrl">
6     <input label="Status" name="okp:doctrl.status" readonly="true"/>
7     <input label="Control" name="okp:doctrl.control" readonly="true"/>
8     <input label="Revision" name="okp:doctrl.revision" readonly="true"/>
9     <input label="Approved By" name="okp:doctrl.approved_by" readonly="true"/>
10    <input label="Approval Date" name="okp:doctrl.approval_date" type="date" readonly="true"/>
11    <input label="Release Date" name="okp:doctrl.release_date" type="date" readonly="true"/>
12  </property-group>
13  <property-group label="Consulting" name="okg:consulting">
14    <input label="Name" type="text" name="okp:consulting.name"/>
15    <input label="Date" type="date" name="okp:consulting.date"/>
16    <checkbox label="Important" name="okp:consulting.important"/>
17    <textarea label="Comment" name="okp:consulting.comment"/>
18  </property-group>
19  <property-group label="Sample Template" name="okg:tpl">
20    <input label="Name" name="okp:tpl.name"/>
21    <input label="Bird Date" name="okp:tpl.bird_date" type="date"/>
22    <select label="Language" name="okp:tpl.language" type="simple">
23      <option label="Java" value="java"/>
24      <option label="Python" value="python"/>
25      <option label="PHP" value="php"/>
26    </select>
27  </property-group>
28 </property-groups>
29

```


The users can assign a property group to a document using these toolbar buttons:

- Add property group → 
- Remove property group → 

And the properties will be shown in the information area. You can edit the values of the properties using the Change button. Then you can change these values. You can also remove a property group from a document by clicking the Delete button.



The screenshot shows a window with tabs: Properties, Notes, History, Security, and Technology. The Technology tab is active. Below the tabs are buttons for 'Change' and 'Delete'. The main area contains the following fields:

- Comment:** A text box containing 'This is a comment'.
- Type:** A dropdown menu showing 'Type One'.
- Language:** A text box containing 'Java'.
- Description:** A text box containing 'qwerty'.

OpenKM 5.1

To try this, login as okmAdmin and go to Administration → Utilities → Register property groups. Set full path with the location of the local file *PropertyGroups.cnd* and click on the register button. The properties will be registered.



The screenshot shows a dialog titled 'Register property groups'. It contains a text field labeled 'Property Group definition path' with the value '/home/openkm/PropertyGroups.cnd'. To the right of the text field is a button labeled 'Register'.



REGISTERED GROUPS CANNOT NOT BE UNREGISTERED. They can always be updated but they cannot be deleted after registration !!! Try it out on a test installation first.

OpenKM 4.0 and older

You need to create some file, for example **PropertyGroup.cnd** and insert some values in two other files **PropertyGroupBundle.properties** and **PropertyGroupValues.properties**.

- \$JBOSS_HOME/PropertyGroup.cnd
- \$JBOSS_HOME/PropertyGroupBundle.properties
- \$JBOSS_HOME/PropertyGroupValues.properties

About files:

- **PropertyGroup.cnd:** There are general definitions on each parameter (for example the parameter okp:technology.type)

```
<okm = 'http://www.openkm.org/1.0'>
<okg = 'http://www.openkm.org/group/1.0'>
<okp = 'http://www.openkm.org/group/property/1.0'>

[okg:technology] mixin
- okp:technology.type (string) = '' multiple autocreated
- okp:technology.language (string) = '' autocreated
```

```
- okp:technology.description (string) = '' autocreated
- okp:technology.comment (string) = '' autocreated

[okg:consulting] mixin
- okp:consulting.type (string) = '' autocreated
- okp:consulting.comment (string) = '' autocreated
```

- **PropertyGroupValues.properties:** In this file you can see the property values, for example for parameter (okp):technology.type=4,one,two,three) has 3 values (one, two, three) the first number indicates the order, the second one the type (1 = list, 2 = input, 3 = select multiple, 4 = Text area)

```
okp\:technology.type=1,4,one,two,three
okp\:technology.language=2,3,java,c,c#,c++,python
okp\:technology.description=3,1
okp\:technology.comment=4,2
okp\:consulting.type=1,3,one,two,three
okp\:consulting.comment=2,2
```

- **PropertyGroupBundle.properties:** This file contains general translations for each value. You can have more files for each language like PropertyGroupBundle_ES.properties etc.

```
okg\:technology=Technology
okp\:technology.type=Type
okp\:technology.type.one=Type One
okp\:technology.type.two=Type Two
okp\:technology.type.three=Type Three
okp\:technology.language=Language
okp\:technology.language.java=Java
okp\:technology.language.c=C
okp\:technology.language.c++=C++
okp\:technology.language.python=Python
okp\:tecnology.description=Description
okp\:technology.comment=Comment
okg\:consulting=Consulting
okp\:consulting.type=Type
okp\:consulting.type.one=Type One
okp\:consulting.type.two=Type Two
okp\:consulting.type.three=Type Three
okp\:consulting.comment=Comment
```


OpenKM tries to find the correct language in your property files. If it does not exist it uses *PropertyBundle.properties*.

Logged users

In OpenKM logged users administrator can see logged users. It's shown the user token (unique session identifier), the datetime when user has logged and the last user datetime.

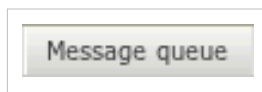
Logged users 					
User	Session id	Remote IP	Remote host	Creation	Last accessed
okmAdmin	94202D14CB549AE2765B008BC9E6020D	127.0.0.1	localhost	19-oct-2010 12:25:04	19-oct-2010 12:38:41

Up to version 4.x and earlier


Uses can be logout. To logout a user click in  logout user icon



Message queue

In Logged_users there's available option **Message queue**

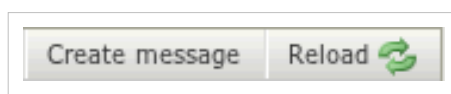


In message queue list it's shown the messages. Messages could be **Permanent** or **Temporal**. **Permanent** messages should be used to advice in time for some event, for example a scheduled maintenance of application several days before starting. **Temporal** messages should be used to advice connected users, for example logout in 5 minutes action.

If **Show** option is not checked - default - when user receives some notification, only will be shown on top left the new notification icon  otherside if shown option is selected, automatically when user receives new notification the popup is shown.

Date	action	message	Type	Show	
Feb 02, 2012 10:15:05 AM		Maintenance advisor for 21-01-2012	Permanent	<input checked="" type="checkbox"/>	 

At top right there're two available options **Create message** and **Reload**.



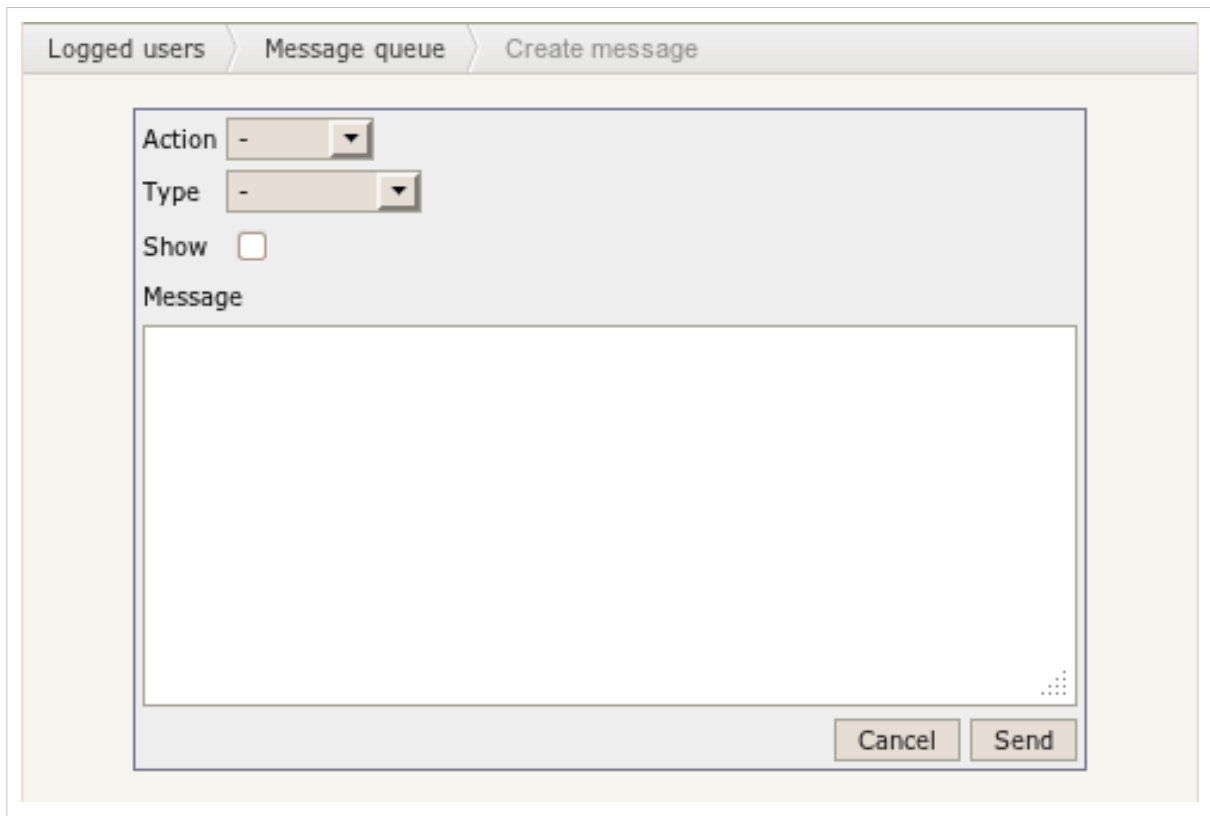
Reload

Reload option will refreshing message queue list. Temporal messages will be automatically deleted.

Create message

Selecting this option can add new message to queue.

- **Action** indicates if must be automatically executed some action associated to message.
- **Type** indicates if message is **Temporal** or **Permanent**.
- **Show** indicates if must be shown automatically the popup from end user UI.




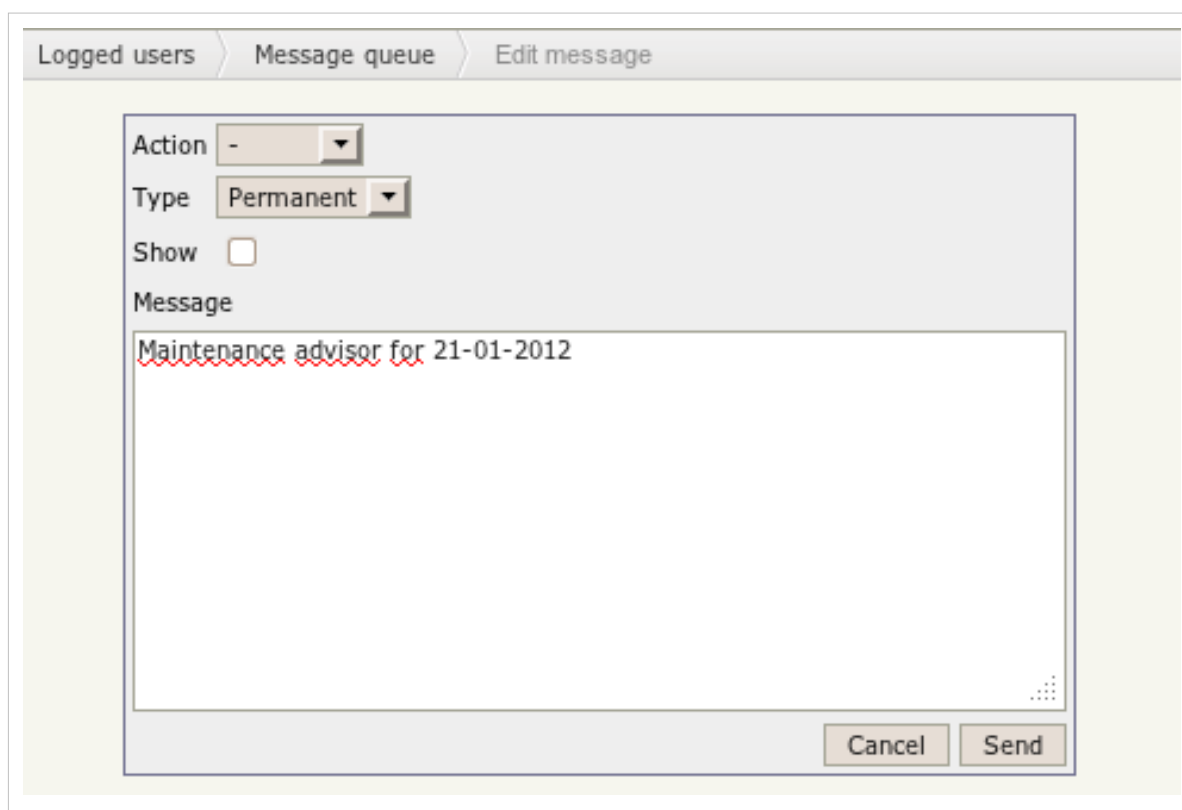
The screenshot shows a web application interface with three tabs: 'Logged users', 'Message queue', and 'Create message'. The 'Create message' tab is active. Inside this tab, there is a form with the following fields:

- Action**: A dropdown menu with a '-' sign.
- Type**: A dropdown menu with a '-' sign.
- Show**: A checkbox.
- Message**: A large text area for entering the message content.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Send'.

Edit message

Click on edit icon  will be shown edit message view. Change the values and click on **Send** button.



Logged users Message queue Edit message

Action -

Type Permanent


Show ☐

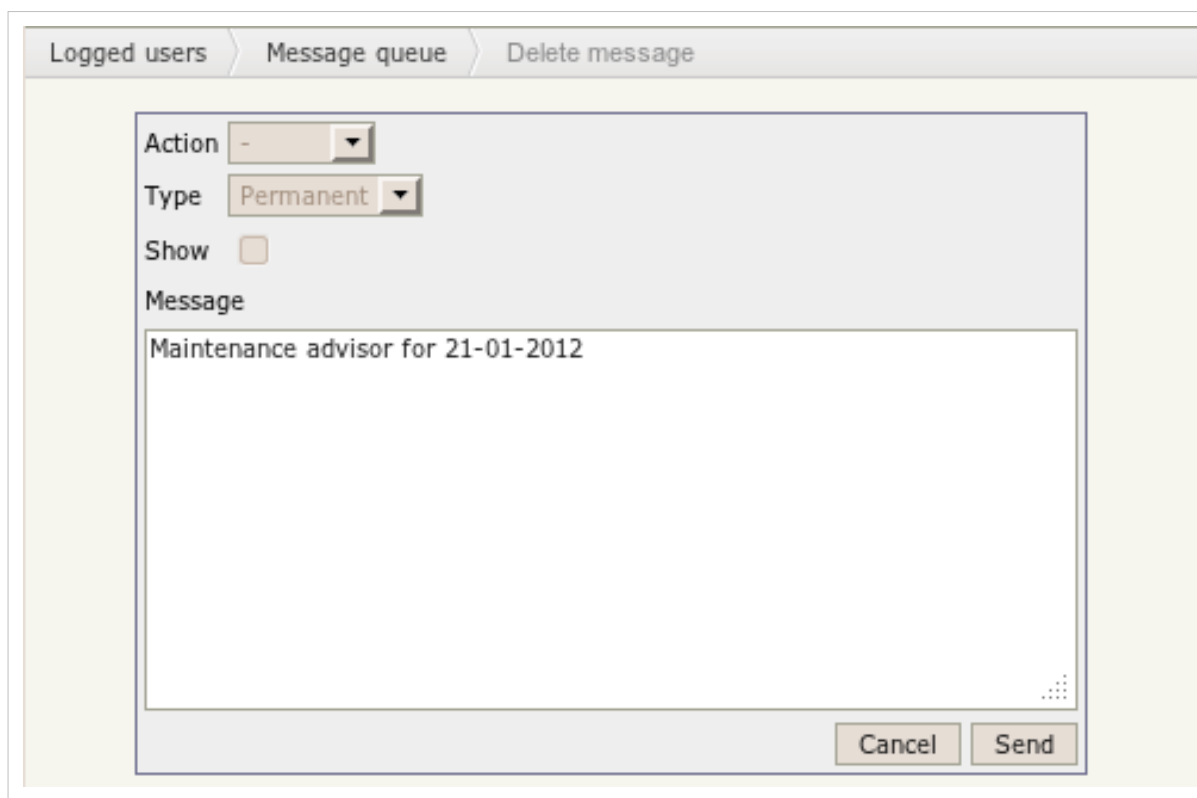
Message

Maintenance advisor for 21-01-2012

Cancel Send

Delete message

Click on delete icon  will be shown delete message view. Click on **Send** button to delete the message.



Logged users Message queue Delete message

Action -

Type Permanent

Show ☐

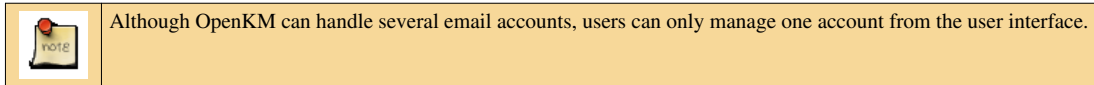
Message

Maintenance advisor for 21-01-2012

Cancel Send

User administration

With User administration view you can **create**, **update**, **delete**, **enable** and **disable** users accounts and **assign** several mail account address and **twitter** accounts.



Detail of the functions according to their icons:

- → Create new user
- → User enabled
- → User disabled
- → Edit user
- → Delete user
- → Edit mail accounts
- → Edit twitter account

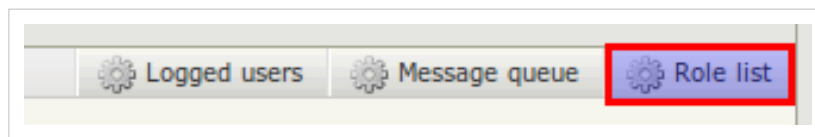
In main user administration view, you can take a quick look at actual users.

Role

#	Id	Name	Mail	Roles	Profile	Active	Chat	
1	okmAdmin	Administrator	noreply@openkm.com	ROLE_ADMIN	Default			
2	sample	Sample User	sample@openkm.com	ROLE_USER				

Switch User / roles view

In the top window you can switch between users and roles view list by clicking on the proper option














User view


Role

#	Id	Name	Mail	Roles	Profile	Active	Chat	
1	okmAdmin	Administrator	noreply@openkm.com	ROLE_ADMIN	Default			
2	sample	Sample User	sample@openkm.com	ROLE_USER				

Role view

#	Id	Active			
1	ROLE_ADMIN				
2	ROLE_USER				

Creating a new role

To create a new role click on the  create new role icon:

Create role


Id

Cancel

Send

After creating the new role, you'll return to view all roles list (now with the role you've created).

Deleting a role

To delete a role click on the  delete role icon:

Delete role

Id

Cancel

Send


After deleting a role, you'll return to view all roles list.

Filtering by role

Using filtering form, you can select to filter the list only by users with some Role.















Role

ROLE_ADMIN




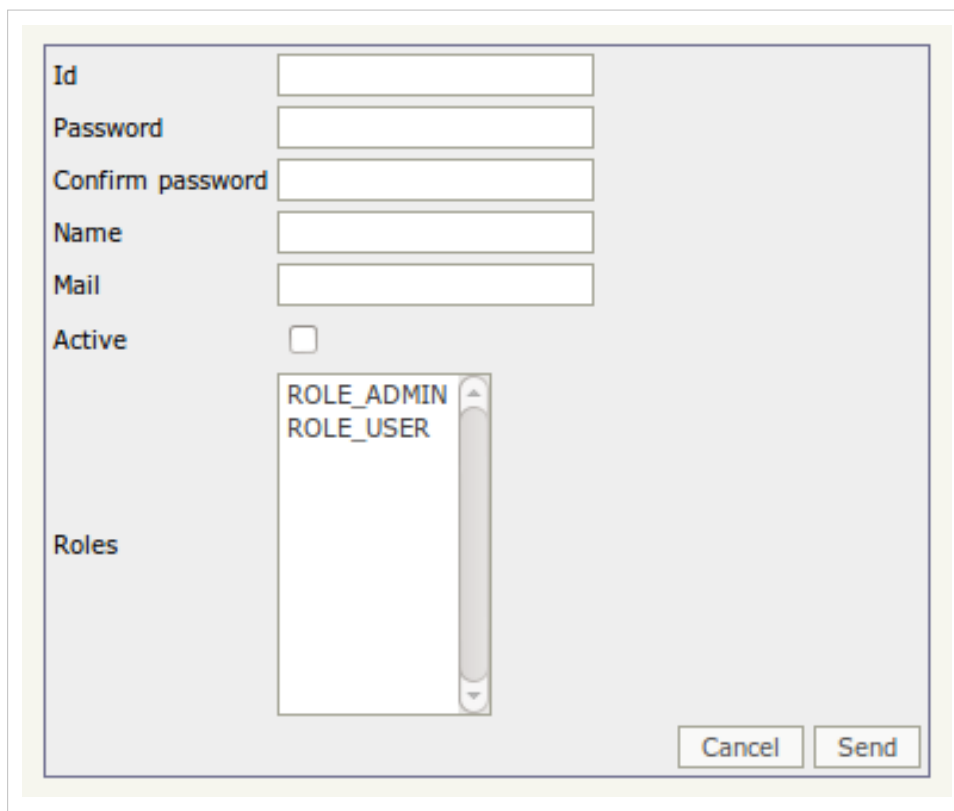
Search

Force user mail import

#	Id	Name	Mail	Roles	Profile	Active	Chat						
1	okmAdmin	Administrator	noreply@openkm.com	ROLE_ADMIN	Default								

Creating a new user

To create a new user click on the  **create new user icon**:



The screenshot shows a web form for creating a new user. The form is enclosed in a light gray border. It contains the following elements:


- Id**: A text input field.
- Password**: A text input field.
- Confirm password**: A text input field.
- Name**: A text input field.
- Mail**: A text input field.
- Active**: A checkbox.
- Roles**: A list box containing two items: **ROLE_ADMIN** and **ROLE_USER**.
- Buttons**: Two buttons at the bottom right, labeled **Cancel** and **Send**.

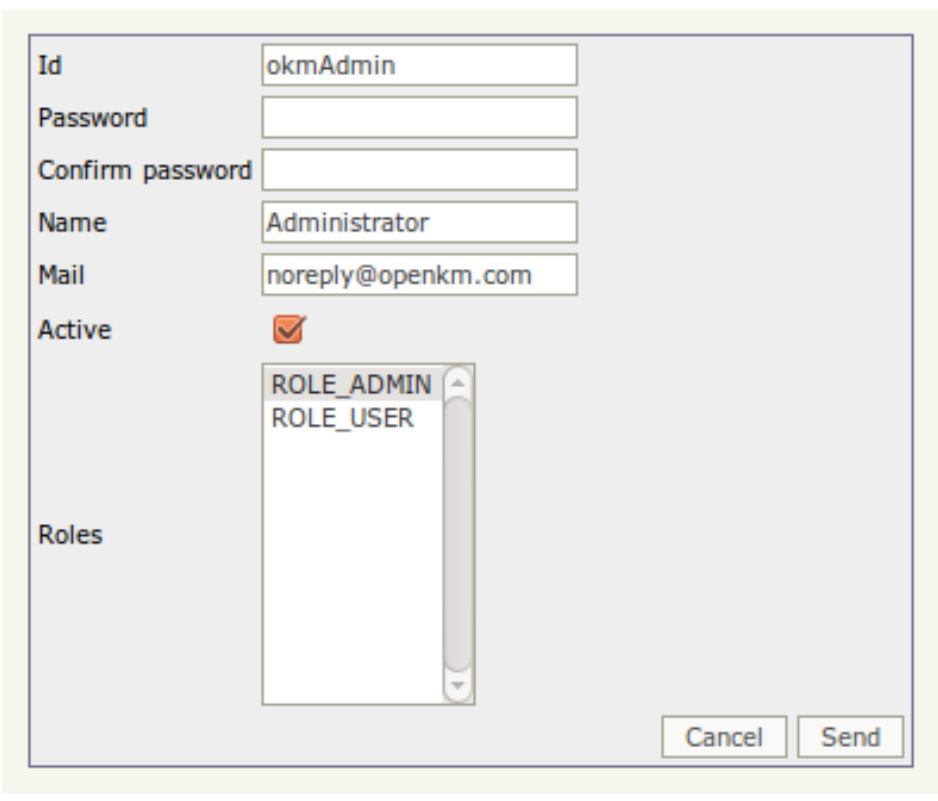
After creating the new user, you'll return to view all users list (now with the user you've created).



Remember that every user should have assigned the **ROLE_USER** or **ROLE_ADMIN** to log into OpenKM. In OpenKM 5.1 the roles where *UserRole* or *AdminRole*.

Updating a user

To update an user have to click in  **edit user icon**:




A screenshot of a web form for updating a user. The form is enclosed in a light gray border. It contains several input fields and a checkbox. The fields are labeled on the left and have their current values in the input boxes on the right. The 'Id' field contains 'okmAdmin'. The 'Password' and 'Confirm password' fields are empty. The 'Name' field contains 'Administrator'. The 'Mail' field contains 'noreply@openkm.com'. The 'Active' field has a checked checkbox. Below the 'Active' field is a scrollable list box labeled 'Roles' containing 'ROLE_ADMIN' and 'ROLE_USER'. At the bottom right of the form are two buttons: 'Cancel' and 'Send'.

Id	okmAdmin
Password	
Confirm password	
Name	Administrator
Mail	noreply@openkm.com
Active	<input checked="" type="checkbox"/>
Roles	<div>ROLE_ADMIN ROLE_USER</div>

Cancel Send

After updating a user, you'll return to view all users list.



Remeber to activate the mail import scheduler. Read more at [Configuring schedulers](#)^[1].

Creating user mail accounts

To insert a new mail user account click on the  **new mail account icon**:

Create mail account

Mail protocol

pop3

Mail host

somehost.com

Mail user

someuser

Mail password

....

Mail folder

Mail mark seen

☐

Mail mark deleted

☐

Active


☐

Check

Cancel



Send


In the page you should set some mail account information like mail protocol, user mail, host server, mail server password IMAP folder and if when mail is read it should be marked as read or to be deleted.



In Google Mail, these IMAP folders are called labels.


Mail accounts (Users)

OKM user	Mail host	Mail user	Mail password	Mail folder	Active	
test	host.com	test	test	imap_folder	true	 






OpenKM periodically will import mails from IMAP pop3 or IMAPS accounts.

Deleting mail accounts


To delete a user related mail account, click on the  **edit mail accounts icon**:


Mail accounts (Users)

OKM user	Mail host	Mail user	Mail password	Mail folder	Active	
test	host.com	test	test	imap_folder	true	 

Then click on  **Delete mail account**. After deleting the mail account, you'll return to view all mail account list.

Updating mail accounts

To update a user mail account, first click on the  **edit mail accounts icon**

Mail accounts (Users)						
OKM user	Mail host	Mail user	Mail password	Mail folder	Active	
test	host.com	test	test	imap_folder	true	

Then click on  **Edit mail account**

Edit mail account


Mail protocol	pop3
Mail host	somehost.com
Mail user	someuser
Mail password	••••
Mail folder	
Mail mark seen	<input type="checkbox"/>
Mail mark deleted	<input type="checkbox"/>
Active	<input type="checkbox"/>

CheckCancelSend

After updating the mail account, you'll return to view all mail account list.

Configuring mail filters

Then click on  **Filter mail account**

Mail filters (Mail accounts)			
Folder	Grouping	Active	
			

Then click on  **Add new mail filter**

Must select a valid OpenKM folder where mails will be imported. If the grouping option is selected, the mail will be imported into a folder structure based on year/date/day of the mail

Create mail filter

Folder

Grouping ☒

Active ☐

Cancel Send

Add mail filtering rules

Then click on  **Rule mail filter**

Filters rules (Mail filters)

Field	Operation	Value	Active	
				

Then click on  **Add new rule**

Create filter rule

Field

Operation

Value

Active ☐


Cancel Send

It must be selected the field (from, to subject content) that must have the valid condition (contains / equals) with some value. Note if you've got several rules in a filter rules, all must be true.


Filters rules (Mail filters)

Field	Operation	Value	Active	
FROM	CONTAINS	project	✗	 

User twitter accounts

To view the user twitter accounts, click on the  **edit twitter accounts icon**:

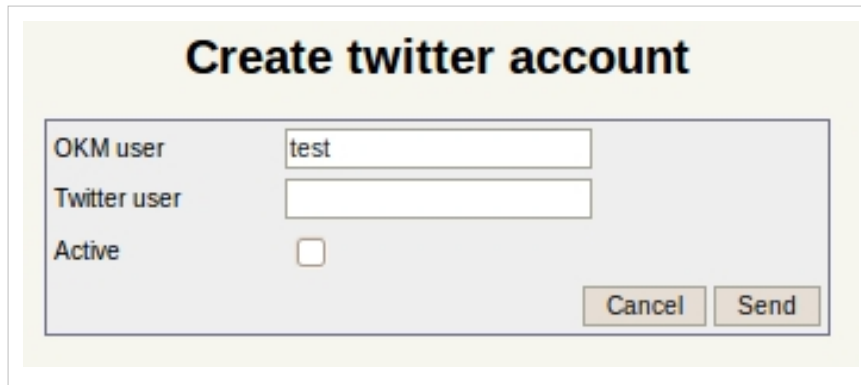
Twitter accounts (Users)

OKM user	Twitter user	Active	
			

There will appear a list of all twitter accounts assigned to a user.



Creating user twitter accounts


To insert a new twitter user account click on the  **new twitter account icon**:



The form is titled "Create twitter account". It contains three input fields: "OKM user" with the value "test", "Twitter user" which is empty, and "Active" which is a checkbox. At the bottom right are "Cancel" and "Send" buttons.


In the following screen, set some twitter account information like twitter user.


Twitter accounts (Users)			
OKM user	Twitter user	Active	
test	test	true	 



OpenKM can be configured to send user notification with twitter.

Deleting twitter accounts


To delete a user's twitter accounts, first click on the  **edit twitter accounts icon**



Twitter accounts (Users)			
OKM user	Twitter user	Active	
test	test	true	 

Then click on  **Delete twitter account**

After deleting the twitter account, you'll return to view all twitter account list.

Updating twitter accounts

To update a user's twitter accounts, first click on the  **edit twitter accounts icon**

Twitter accounts (Users)			
OKM user	Twitter user	Active	
test	test	true	 

Then click on  **Edit twitter account**



Update twitter account

OKM user

Twitter user

Active ☒

After updating the twitter account, you'll return to view all twitter account list.

References

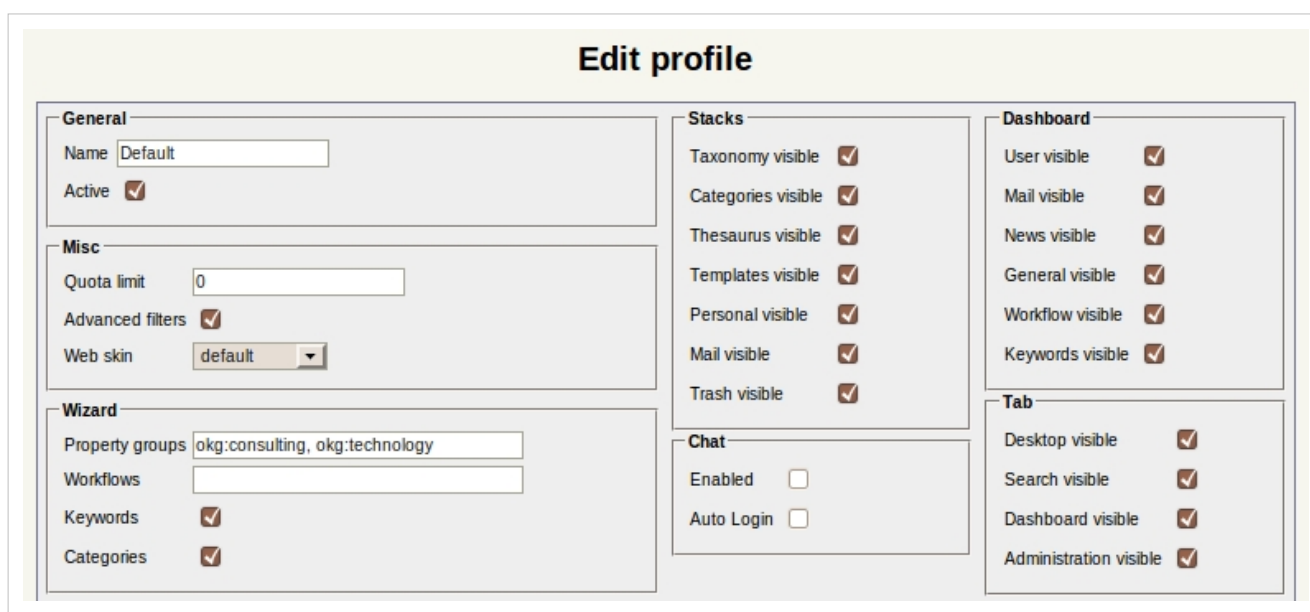
[1] http://wiki.openkm.com/index.php/OpenKM.cfg#Configuring_schedulers

Profiles

Each user must be assigned a profile. A profile is a definition of general configuration parameters of the UI interface, available extensions, and other OpenKM features that allows for customization of the end user interface.



Name	Active	
Default	<input checked="" type="checkbox"/>	 



Edit profile

General

Name

Active ☒

Misc

Quota limit

Advanced filters ☒

Web skin

Wizard

Property groups

Workflows

Keywords ☒

Categories ☒

Stacks

Taxonomy visible ☒

Categories visible ☒

Thesaurus visible ☒

Templates visible ☒

Personal visible ☒

Mail visible ☒

Trash visible ☒

Chat

Enabled ☐

Auto Login ☐

Dashboard

User visible ☒

Mail visible ☒

News visible ☒

General visible ☒

Workflow visible ☒

Keywords visible ☒

Tab

Desktop visible ☒

Search visible ☒

Dashboard visible ☒

Administration visible ☒

Special parameters

Quota limit allow users to upload documents up to some size. This is useful for the administrator to take better control of repository growth, and detecting application abuse by users.

Advanced filters should be used when the number of users is high. This enables the user and group filtering features.

Web skin is useful for cyrilic and similar languages like chinese, japanese etc... For these languages it is useful to use **medium font** by default.

Wizard **property group** will make OpenKM demand for some metadata each time a new file is uploaded. The metadata group names are put separated by ","

Wizard **workflow** will make OpenKM demand for some workflow each time a new file is uploaded. The metadata group names are put separated by ","

Wizard **keyword** will make OpenKM demand for some keywords each time a new file is uploaded.

Wizard **categories** will make OpenKM demand for some categories each time a new file is uploaded.



If some property wizard is enabled, any new document uploaded will demand some extra information. Wizard properties are for the whole repository without distinctions.

Add new profile

Then click on the **new profile icon**

Create profile

General	Stacks	Dashboard
Name <input type="text"/>	Taxonomy visible <input type="checkbox"/>	User visible <input type="checkbox"/>
Active <input type="checkbox"/>	Categories visible <input type="checkbox"/>	Mail visible <input type="checkbox"/>
	Thesaurus visible <input type="checkbox"/>	News visible <input type="checkbox"/>

Edit a profile

Then click on the **edit profile icon**

Edit profile

General	Stacks	Dashboard
Name <input type="text" value="Default"/>	Taxonomy visible <input checked="" type="checkbox"/>	User visible <input checked="" type="checkbox"/>
Active <input checked="" type="checkbox"/>	Categories visible <input checked="" type="checkbox"/>	Mail visible <input checked="" type="checkbox"/>

Database query

Using database query, administrators can make queries on OpenKM databases. The query type can be JDBC or Hibernate.

Database query

select * from okm_user

JDBC
Send

USR_ID	USR_NAME	USR_PASSWORD	USR_EMAIL	USR_ACTIVE
josep	josep	ea51218d782eb259bf2d3d61a48ae8a8	none@none.com	true
okmAdmin	Administrator	21232f297a57a5a743894a0e4a801fc3	admin@noreply.com	true

Improvements from OpenKM 5.1.6:

- Several types of queries: JDBC, Hibernate and Metadata.
- If you select JDBC, a list of tables will be shown.
- If you select Metadata, a list of meta-tables will be shown.

See also Database Metadata.

Reports

In the report section you can create and use your own Jasper Reports. Reports are generated in PDF format. Starting from OpenKM 5.1, other formats can be generated.

The JasperReports version included in OpenKM 6.2 works best with [iReport 3.7.6.^[1]].

Reports

Name	Type	File Name	Active	
				+

Reports usually make use of certain TTF fonts that need to be installed in your server. For example, the Arial font. These fonts are included in Windows but not in Linux distros. To install these fonts under Debian / Ubuntu:


```
$ sudo aptitude install msttcorefonts
```

To install under CentOS run these commands as root:

```
$ yum install cabextract rpm-build
$ wget http://corefonts.sourceforge.net/msttcorefonts-2.0-1.spec
$ rpmbuild -ba msttcorefonts-2.0-1.spec
$ rpm -ivh /usr/src/redhat/RPMS/noarch/msttcorefonts-2.0-1.noarch.rpm
$ /sbin/service xfs reload
```

Add new report

Then click on the  new report icon



The 'Create report' dialog box contains the following fields and controls:

- Name:** A text input field.
- Type:** A dropdown menu currently showing 'SQL'.
- Active:** An unchecked checkbox.
- File:** A text input field with an 'Examiner...' button to its right.
- Buttons:** 'Cancel' and 'Send' buttons at the bottom right.

Reports can use internal Scripting or SQL. The file to upload must be a **.jrxml** JasperReport source file. Starting with OpenKM 5.1 you can also upload compiled reports (files with **.jasper** extension).

Types of reports

SQL Report

Usually you have a database and want to generate a report extracting data from it. In this case you should know the table relationships. Tools like Aqua Data Studio ^[2] can help you to create a ER diagram. In this example, we will query the database to generate a report with information related to workflow execution:

```
select ti.actorid_ as ti_actorid, pd.name_ as pd_name, ti.name_ as
ti_name, ti.start_ as ti_start, ti.end_ as ti_end
from JBPM_TASKINSTANCE as ti, JBPM_PROCESSINSTANCE as pi,
JBPM_PROCESSDEFINITION as pd
where ti.procinst_ = pi.id_ and pi.processdefinition_ = pd.id_
order by ti_actorid, pd_name, ti_name, ti_start, ti_end
```

Full report is available at .

Script Report

Other times, the data is not extracted from a database but another source. In this case we will query the OpenKM API to obtain a list of registered users.

```
import com.openkm.dao.*;
import com.openkm.dao.bean.*;

List al = new ArrayList();
for (User user : AuthDAO.findAllUsers(false)) {
    Map usr = new HashMap();
    usr.put("id", user.getId());
    usr.put("name", user.getName());
    usr.put("email", user.getEmail());
    usr.put("roles", user.getRoles().toString());
    al.add(usr);
}
```

```

}

return al;

```

This means that not only database-managed users are shown but also external users if you have configured OpenKM to use an external LDAP server to handle user management.

Full report is available at .

XPath Report



This feature is only available in OpenKM 5.1.

If you want to create a report which uses an XPath query, use scripting. This code will query for locked documents:

```

import javax.jcr.*;
import javax.jcr.query.*;
import org.apache.jackrabbit.*;
import com.openkm.module.direct.*;

List al = new ArrayList ();
String statement =
"/jcr:root/okm:root//element(*,okm:document) [@jcr:lockOwner]/@jcr:lockOwner";
String type = "xpath";
Session jcrSession = DirectRepositoryModule.getSession();
Workspace workspace = jcrSession.getWorkspace();
QueryManager queryManager = workspace.getQueryManager();
Query query = queryManager.createQuery(statement, type);
QueryResult result = query.execute();

for (RowIterator it = result.getRows(); it.hasNext();) {
    Map ld = new HashMap();
    javax.jcr.query.Row row = it.nextRow();
    javax.jcr.Value v = row.getValue(JcrConstants.JCR_LOCKOWNER);
    ld.put("owner", v==null?"NULL":v.getString());
    v = row.getValue(JcrConstants.JCR_PATH);
    ld.put("path", v==null?"NULL":v.getString());
    al.add(ld);
}

return al;

```

You can find the whole report at and .

Reports with parameters

Since OpenKM 5.1.7, the reports feature has been improved and now you can create reports with parameters. This enables the possibility of creating more dynamic reports.

To achieve this you need to package the report in an archive with a **rep** extension. This archive will contain the report itself and an XML file called **params.xml** which describes the report parameters. This file must be called **params.xml** or this won't work. You can create a ZIP archive from these files and rename the extension to "rep".

Let's see a sample **params.xml** file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report-parameters PUBLIC "-//OpenKM//DTD Report Parameters 2.0//EN"
"http://www.openkm.com/dtd/report-parameters-2.0.dtd">
<report-parameters>
  <input label="Path" name="path" type="folder"/>
</report-parameters>
```

In this XML you can see an "input" parameter called "path". The "type" field specifies what kind of "input" will be used (for more info read Report Parameters description) in this piece of the script inside the report. Let's see how this parameter can be used:

```
List al = new ArrayList ();
Session jcrSession = DirectRepositoryModule.getSession();
System.out.println("Parameter 'path': " + path);
Node root = jcrSession.getRootNode().getNode(path.substring(1));
nodeIterator(root);
```

As you can see, the parameter "path" is automatically injected and can be used as a local variable.


You can download the sample report at .

















References

- [1] http://jasperforge.org/plugins/espnews/browse.php?group_id=83&news_id=256
- [2] <http://www.aquafold.com/>

Activity log

Any operation done by OpenKM users are automatically audited in the database. This is a complete audit tracking log for any operation done in OpenKM.

OpenKM is 100% compliant with data protection law.



Activity log

From

To

User

Action

Send

You must select a date range

For example we can take a look at any document created by users (among other available operations) as can be seen in the next screenshot.

Activity log

From

To

User

Action

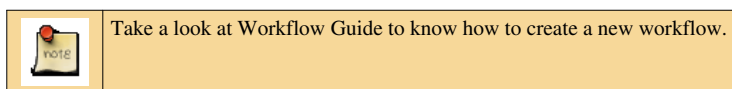
Send

Date	User	Token	Action	Item	Params
Wed Mar 03 17:38:15 CET 2010	okmAdmin	83554556567289642956189114652	CREATE_DOCUMENT	/okm:root/test/paco.txt	text/plain, 1404
Wed Mar 03 17:47:52 CET 2010	okmAdmin	83554556567311752956194845812	CREATE_DOCUMENT	/okm:root/test/paco.txt	text/plain, 1404
Thu Mar 04 17:32:12 CET 2010	okmAdmin	83554556570482262957006574091	CREATE_DOCUMENT	/okm:root/marketing/paco.txt	text/plain, 1404
Mon Mar 15 17:52:11 CET 2010	okmAdmin	835545566078218529516565423342	CREATE_DOCUMENT	/okm:root/test/paco.txt	text/plain, 1404
Wed Mar 17 13:24:30 CET 2010	okmAdmin	835545566139435929518132610451	CREATE_DOCUMENT	/okm:root/motoluis_pendent.txt	text/plain, 2042
Thu Mar 18 17:49:46 CET 2010	okmAdmin	83554556617727929519155015642	CREATE_DOCUMENT	/okm:root/plantillas.txt	text/plain, 582

Workflow

In workflow view, the administrator can add, delete or update workflow definitions. Administrators can see the workflow process and status.

In this view, you can upload a new workflow definition (.par file).



Process Definitions

Process ID	Process Name	Version	Actions
4	advanced	1	
3	caracas	2	
2	caracas	1	
1	simple	1	

Upload process definition

The workflow status and the processes that are running can be seen.

Process Definition

Process ID	Name	Description	Version
3	caracas		2

Process Instances

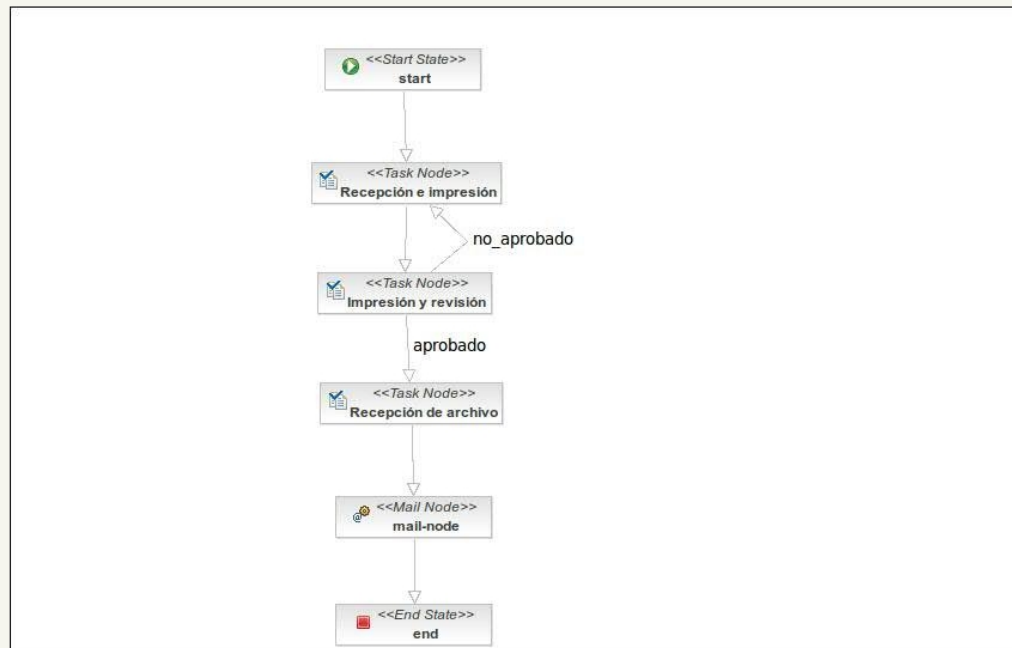
Instance ID	Key	Status	Start Date	End Date	Actions
11		Running	Thu Mar 25 16:45:55 CET 2010		
10		Running	Thu Mar 25 16:29:11 CET 2010		
9		Running	Thu Mar 25 16:28:10 CET 2010		
8		Running	Thu Mar 25 15:29:29 CET 2010		
7		Running	Wed Mar 24 11:23:17 CET 2010		

Forms

Task	Form						
	Label	Name	Value	Width	Height	Field	Others
imprimir	Documento impreso			100px	25px	Button	Type: transition
archivar	Documento archivado			100px	25px	Button	Type: transition
revisar	Motivo desaprovação	motivo		300px	100px	TextArea	
	Aprobar	aprobado		100px	25px	Button	Type: transition
	No aprobar	no_aprobado		100px	25px	Button	Type: transition

And see the graphical representation of the workflow .

Process Image



Each process can be edited to change it, assign variables, modify some data, and see it's actual state etc...

Process Instance

Instance ID	Key	Process	Status	Start Date	End Date
11		caracas v2	Running	Thu Mar 25 16:45:55 CET 2010	

Tasks Instances

ID	Name	Pooled Actors	Assigned To	Status	Start Date	End Date	Actions
17	imprimir		dcentral	Ended	Thu Mar 25 16:45:55 CET 2010	Thu Mar 25 16:46:18 CET 2010	
18	revisar		pgeneral	Ended		Thu Mar 25 16:50:04 CET 2010	
19	imprimir		dcentral	Running	Thu Mar 25 16:50:04 CET 2010		

Comments

Actor ID	Time	Comment
dcentral	Fri Mar 26 13:07:53 CET 2010	
dcentral	Fri Mar 26 13:08:02 CET 2010	test de nota que se ha añadido
dcentral	Fri Mar 26 13:20:55 CET 2010	test de cmentari
dcentral	Fri Mar 26 13:22:58 CET 2010	manolo
dcentral	Fri Mar 26 13:24:35 CET 2010	alegria
dcentral	Fri Mar 26 13:24:45 CET 2010	alegria 2

Add comment

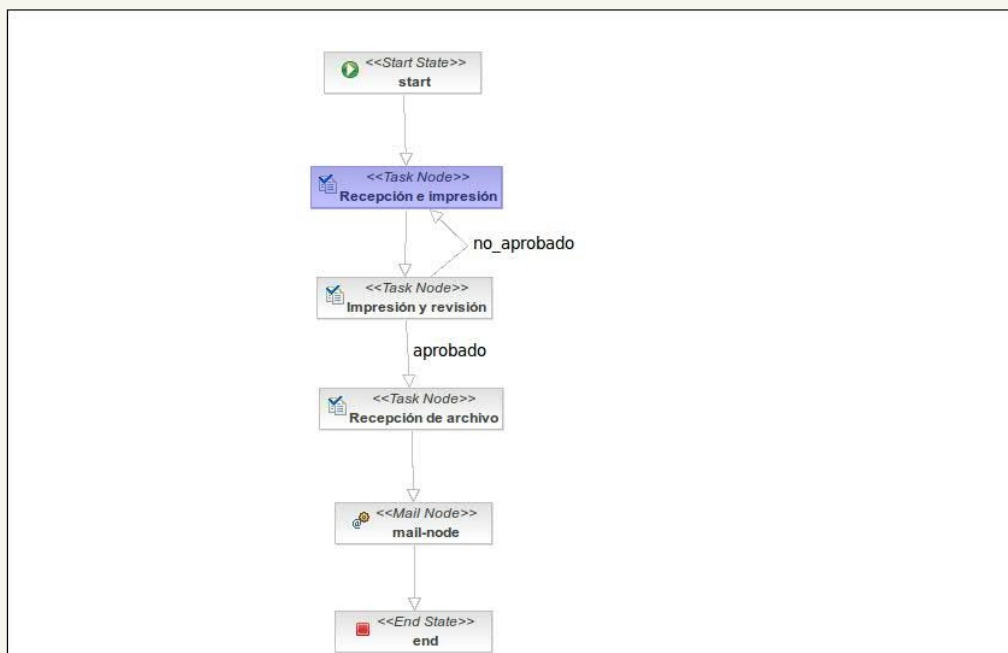
Tokens

Token ID	Parent	Node	Status	Start Date	End Date	Actions
11	(no parent)	Recepción e impresión	Running	Thu Mar 25 16:45:55 CET 2010		

Process Variables

Name	Value	Actions
motivo	este es el motivo	
path	/okm:root/plantillas.txt	

Process Image



Automation



Since OpenKM 6.4.2 the plug-in system of OpenKM allows you to quickly expand the functionality offered by the platform, extending the available Automation without having to rebuild the system to add/change the existing functionality. Refer to Extension Guide if you need to extend automation feature on 6.4.2.

For OpenKM version older than 6.4 refer to Enable automation section to know how enable automation. By default in these versions automation is not enabled.


Refer to Extend automation to extend automation validators and actions.

Order	Name	Event	At	Validations	Actions	Exclusive	Active	
1	Add keyword	doc_move	post	0	0	✗	✓	

Preliminar concepts

- **Rule** is composed by validators and actions.
- **Validators** are one or more conditions on a rule that are evaluated depending the elements which fired the automation event (document, folder, etc...)
- **Actions** are some operations with element which fired the automation event (document, folder, etc...)
- **Rule is executed** when all validators in a rule return true.
- When **Actions are executed** (all validators in rule are true) all actions defined in the rule are executed in the element which fired the automation event (folder, document, etc...)

Add new rule

Click  icon and will be shown the add rule form:

Order

1

Name

Rule to add automatic category

Event

doc_move

At

post

Exclusive

☐

Active




















☒







Cancel

Send


- **Order** field indicates execution preference among the other rules.
- **Name** field is some name to identify the rule.
- **Event** field indicates on which type of events is evaluated the rule.
- **At** field has two possible values post or pre. **post** indicates execution before element is created at OpenKM, **pre** indicates execution when element is yet created. With example will be more clear, with **post** we can change the path where a document will be created, with **pre** the document is yet created but we can move to other path.
- **Exclusive** field enabled stop the evaluation is rule is executed. Automation can have several rules, one or more can be evaluated and executed at same time (in order defined by field **Order**), but if a rule with exclusive field enabled is executed then automation stops here (it's similar concept than firewall rules).
- **Active** indicates if rule is enabled or disabled.

Click **send** button and the rule will be added.



Order	Name	Event	At	Validations	Actions	Exclusive	Active	
1	Add keyword	doc_move	post	0	0	✗	✓	  
1	Rule to add automatic category	doc_move	post	0	0	✗	✓	  

Edit a rule

Click  icon and will be shown the edit rule form:

Order

1

Name

Rule to add automatic category

Event

doc_move

At

post

Exclusive

☐

Active


☒

Cancel

Send

Changes values if needed and then click **send** button to update the rule.

Delete a rule

Click  icon and will be shown the delete rule form:

Order

1

Name

Rule to add automatic category

Event

doc_move

At

post

Exclusive

☐

Active


☒

Cancel

Send

Click **send** button and the rule will be deleted.

Edit actions and validators

Click  icon will be shown a screen like this:

Add validation:

-

+


VALIDATIONS				
Order	Type	active	Param0	Param1

Add action:

-

+

ACTIONS				
Order	Type	active	Param0	Param1

Click  icon and add some validator:

Add validation: PathContains +

Create validation

Name

PathContains

Group

validation

ClassName

com.openkm.automation.validation.PathContains

Active

☒

Parameters:

Order

1

String

/okm:root



Cancel

Send

VALIDATIONS				
Order	Type	active	Param0	Param1
Add action: - +				
ACTIONS				
Order	Type	active	Param0	Param1

Click **send** button and the validator will be added.

Add validation: - +

VALIDATIONS				
Order	Type	active	Param0	Param1
1	PathContains	✓	String:/okm:root	 

Add action: - +

ACTIONS				
Order	Type	active	Param0	Param1

Click + icon and add some action:

Add validation: - +

VALIDATIONS					
Order	Type	active	Param0	Param1	
1	PathContains	✓	String:/okm:root		

Add action: ExecuteScripting +

Create action

Name ExecuteScripting

Group action

ClassName com.openkm.automation.action.ExecuteScripting

Active ☒

Parameters:

Order

Script

ACTIONS					
Order	Type	active	Param0	Param1	

Click **send** button and the action will be added.

Add validation: - +

VALIDATIONS					
Order	Type	active	Param0	Param1	
1	PathContains	✓	String:/okm:root		

Add action: - +

ACTIONS					
Order	Type	active	Param0	Param1	
1	ExecuteScripting	✓			

With edit icon and delete icon you can edit or delete validators and actions.

Validators description

Element can be document, folder, etc...

ContainsFolder

- Return **True** if element path **has some folder child**.
- Example
 - Folder child name = test
 - Element folder with path **/okm:root/folder1** has a folder called **/okm:root/folder1/test** then return true
 - Element folder with path **/okm:root/folder1** has not a folder called **/okm:root/folder1/test** then return false

HasKeyword

- Return **True** if element has **some keyword**

HasPropertyGroup

- Return **True** if element has **some property group**

HasPropertyGroupValue

- Return **True** if element has **some property group value**

NameContains

- Return **True** if path element contains **some string value**
- Example
 - Value = test
 - Element folder with path **/okm:root/folder1/test/temporary** contains **test**, then return true
 - Element folder with path **/okm:root/folder1/temporary** not contains **test**, then return false

PathContains

- Return **True** if path starts **with some value**
 - Path starts with = **/okm:root/folder1**
 - Element folder with path **/okm:root/folder1/test/temporary** return true
 - Element folder with path **/okm:root/test/temporary** returns false

PathContainsFolderName

- Return **True** if path contains **some value**
 - Folder name with = **folder1**
 - Element folder with path **/okm:root/folder1/test/temporary** return true
 - Element folder with path **/okm:root/test/temporary/folder1** return true
 - Element folder with path **/okm:root/test/temporary** returns false

UserHasRole

- Return **True** if user has **some role**

Actions description

Element can be document, folder, etc...

AddCategory

Add some category to the element

AddCategoryToWizard

After uploading document will be shown the wizard (user action needed) to add categories.

AddDigitalSignatureToWizard

After uploading document will be shown the wizard (user action needed) to sign document.

AddKeyword

Add some keyword to the element.

addKeywordToWizard

After uploading document will be shown the wizard (user action needed) to to add keywords.

AddPropertyGroupToWizard

After uploading document will be shown the wizard (user action needed) to to add property groups.

AddWorkflowToWizard

After uploading document will be shown the wizard (user action needed) to to execute workflows.

ExecuteScripting

Execute a script related with the element.

GrantRole

Add some role grant to the element.

GrantUser

Add some user grant to the element.

RevokeAllRoles

Remove all roles grants to the element.

RevokeAllUsers

Remove all user grants to the element.

RevokeCategory

Remove a category to the element.

RevokeKeyword

Remove a keyword to the element.

RevokeRole

Remove a role grant to the element.

RevokeUser

Remove a user grant to the element.

SetTextExtracted

Add text content to be indexed as document content.

SendMail

Send mail to users. Can be used the same variables than Notification_and_subscription_messages. The default subject used is notification subject.

Enable automation



Since OpenKM 6.4.2 the plug-in system of OpenKM allows you to quickly expand the functionality offered by the platform, extending the available Automation without having to rebuild the system to add/change the existing functionality.

OpenKM automation by default is not enabled to enable it you should execute some sql queries.

- Refer to Automation section to configure automation rules in your OpenKM and get full explanation of available validators and actions.
- Refer to Extend automation to extend automation validators and actions.

OpenKM community version

Available validations:

- PathContains

Available actions:

- ExecuteScripting

Enable automation validation

Go to **Administration > Database query** section and execute:

```
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01 ) VALUES
('post', 'com.openkm.automation.validation.PathContains',
'PathContains', 'validation', 'text', 'okm:folder', 'String', '', '',
'');
```

Enable automation actions

Go to **Administration > Database query** section and execute:

```
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01 ) VALUES
('post', 'com.openkm.automation.action.ExecuteScripting',
'ExecuteScripting', 'action', 'textarea', '', 'Script', '', '',
'');
```

OpenKM professional version

Available validations:

- ContainsFolder
- HasKeyword
- HasPropertyGroup
- HasPropertyGroupValue
- NameContains
- PathContains
- PathContainsFolderName
- UserHasRole

Available actions:

- AddCategory
- AddCategoryToWizard
- AddDigitalSignatureToWizard
- AddKeyword
- AddKeywordToWizard
- AddPropertyGroupToWizard
- AddWorkflowToWizard
- ExecuteScripting
- GrantRole
- GrantUser
- RevokeAllRoles
- RevokeAllUsers
- RevokeCategory
- RevokeKeyword
- RevokeRole
- RevokeUser
- SetTextExtracted
- SendMail

Enable automation validation

Go to **Administration > Database query** section and execute:

```
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.ContainsFolder', 'ContainsFolder',
'validation', 'text', '', 'Folder', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.HasKeyword', 'HasKeyword',
'validation', 'text', '', 'Keyword', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.NameContains', 'NameContains',
```



```

'validation', 'text', '', 'String', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.PathContains', 'PathContains',
'validation', 'text', 'okm:folder', 'String', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.HasPropertyGroupValue',
'HasPropertyGroupValue', 'validation', 'text', '', 'Property', 'text',
'', 'Value');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.HasPropertyGroup',
'HasPropertyGroup', 'validation', 'text', '', 'Property group', '', '',
'');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.HasCategory', 'HasCategory',
'validation', 'text', 'okm:folder', 'String', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.PathContainsFolderName',
'PathContainsFolderName', 'validation', 'text', '', 'String', '', '',
'');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.validation.UserHasRole', 'UserHasRole',
'validation', 'text', '', 'Keyword', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('pre',
'com.openkm.automation.validation.UserHasRole', 'UserHasRole',
'validation', 'text', '', 'Keyword', '', '', '');

```

Enable automation actions

Go to **Administration > Database query** section and execute:

```

INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post', 'com.openkm.automation.action.AddKeyword',
'AddKeyword', 'action', 'text', '', 'Keyword', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,

```

```

AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post', 'com.openkm.automation.action.AddCategory',
  'AddCategory', 'action', 'text', 'okm:folder', 'Category', '', '',
  '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('pre',
  'com.openkm.automation.action.AddCategoryToWizard',
  'AddCategoryToWizard', 'action', '', '', '', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
  'com.openkm.automation.action.AddCategoryToWizard',
  'AddCategoryToWizard', 'action', '', '', '', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('pre',
  'com.openkm.automation.action.AddKeywordToWizard',
  'AddKeywordToWizard', 'action', '', '', '', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
  'com.openkm.automation.action.AddKeywordToWizard',
  'AddKeywordToWizard', 'action', '', '', '', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
  'com.openkm.automation.action.AddPropertyGroupToWizard',
  'AddPropertyGroupToWizard', 'action', 'text', '', 'Property group', '',
  '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post', 'com.openkm.automation.action.GrantRole',
  'GrantRole', 'action', 'text', '', 'Role name', 'integer', '',
  'Privileges');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post', 'com.openkm.automation.action.GrantUser',
  'GrantUser', 'action', 'text', '', 'User name', 'integer', '',
  'Privileges');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post', 'com.openkm.automation.action.RevokeRole',
  'RevokeRole', 'action', 'text', '', 'Role name', 'integer', '',
  'Privileges');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,

```

```

AMD_DESC01) VALUES ('post', 'com.openkm.automation.action.RevokeUser',
'RevokeUser', 'action', 'text', '', 'User name', 'integer', '',
'Privileges');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.RevokeKeyword', 'RevokeKeyword',
'action', 'text', '', 'Keyword', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.RevokeCategory', 'RevokeCategory',
'action', 'text', 'okm:folder', 'Category', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.AddWorkflowToWizard',
'AddWorkflowToWizard', 'action', 'text', '', 'Workflow', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.AddDigitalSignatureToWizard',
'AddDigitalSignatureToWizard', 'action', '', '', '', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.SetTextExtracted', 'SetTextExtracted',
'action', 'text', '', 'Text', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.RevokeAllRoles', 'RevokeAllRoles',
'action', 'boolean', '', 'Recursive', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',
'com.openkm.automation.action.RevokeAllUsers', 'RevokeAllUsers',
'action', 'boolean', '', 'Recursive', 'boolean', '', 'Preserve
creator');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('pre',
'com.openkm.automation.action.ExecuteScripting', 'ExecuteScripting',
'action', 'code', '', 'Script', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01) VALUES ('post',

```

```
'com.openkm.automation.action.ExecuteScripting', 'ExecuteScripting',
'action', 'code', '', 'Script', '', '', '');
INSERT INTO OKM_AUTO_METADATA (AMD_AT, AMD_CLASS_NAME, AMD_NAME,
AMD_GROUP, AMD_TYPE00, AMD_SRC00, AMD_DESC00, AMD_TYPE01, AMD_SRC01,
AMD_DESC01, AMD_TYPE02, AMD_SRC02, AMD_DESC02) VALUES ('post',
'com.openkm.automation.action.SendMail', 'SendMail', 'action',
'textarea', '', 'Message', 'user', '', 'Users', 'role', '', 'Roles');
```

Extend automation



From OpenKM 6.4+ the Plug-in system of OpenKM allows you to quickly expand the functionality offered by the platform, extending the available Automation without having to rebuild the system to add/change the existing functionality.

- OpenKM 6.4 (also valid for OpenKM 6.4.+)
- OpenKM 6.0 (also valid for OpenKM 6.X+)

Extend automation 6.4

The Plug-in system of OpenKM allows you to quickly expand the functionality offered by the platform, extending the available Automation without having to rebuild the system to add/change the existing functionality.

Step 1 - Create class

- If you make a new validation the new class should be under `com.openkm.automation.validation` (this is mandatory)
- If you make a new action the new class should be under `com.openkm.automation.action` (this is mandatory)
- **Important** do not forget to set the tag **@PluginImplementation** before the class definition

Validation PathContains example

```
package com.openkm.automation.validation;

import java.util.HashMap;

import net.xeoh.plugins.base.annotations.PluginImplementation;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.openkm.api.OKMRepository;
import com.openkm.dao.bean.Automation;
import com.openkm.automation.AutomationUtils;
import com.openkm.automation.Validation;

/**
 * Check if the current parent path contains a designed one. The only
```

```
* parameter is a path and will test if this one is included in the  
* actual parent.  
*  
*/  
@PluginImplementation  
public class PathContains implements Validation {  
    private static Logger log =  
LoggerFactory.getLogger(PathContains.class);  
  
    @Override  
    public boolean isValid(HashMap<String, Object> env, Object... params) {  
        String uuid = AutomationUtils.getString(0, params);  
        String parentPath = AutomationUtils.getParentPath(env);  
  
        try {  
            String path =  
OKMRepository.getInstance().getNodePath(null, uuid);  
  
            if (parentPath.startsWith(path)) {  
                return true;  
            } else {  
                return false;  
            }  
        } catch (Exception e) {  
            log.error(e.getMessage(), e);  
        }  
  
        return false;  
    }  
  
    @Override  
    public boolean hasPost() {  
        return true;  
    }  
  
    @Override  
    public boolean hasPre() {  
        return false;  
    }  
  
    @Override  
    public String getName() {  
        return "PathContains";  
    }  
  
    @Override  
    public String getParamType00() {
```

```
        return Automation.PARAM_TYPE_TEXT;
    }

    @Override
    public String getParamSrc00() {
        return Automation.PARAM_SOURCE_FOLDER;
    }

    @Override
    public String getParamDesc00() {
        return "String";
    }

    @Override
    public String getParamType01() {
        return Automation.PARAM_TYPE_EMPTY;
    }

    @Override
    public String getParamSrc01() {
        return Automation.PARAM_SOURCE_EMPTY;
    }

    @Override
    public String getParamDesc01() {
        return "";
    }

    @Override
    public String getParamType02() {
        return Automation.PARAM_TYPE_EMPTY;
    }

    @Override
    public String getParamSrc02() {
        return Automation.PARAM_SOURCE_EMPTY;
    }

    @Override
    public String getParamDesc02() {
        return "";
    }
}
```

Step 2 - Publish

Create your own jar and copy into \$TOMCAT_HOME/lib folder.

Extend automation 6.0

Refer to Enable automation section to know how enable automation. By default automation is not enabled.

Refer to Automation section to configure automation rules in your OpenKM and get full explanation of available validators and actions.

Create a new validator

Any validator should implements interface Validation:

```
public interface Validation {
    public static final String METHOD = "isValid";

    boolean isValid(HashMap<String, Object> env, Object... params);
}
```

- At HashMap<String, Object> env are stored element values map.
- Values set at validator form are stored to String params[].



Refer to AutomationUtils class to get variables. Take a look at the latest doxygen class reference at Developer_Guide.

```
public class PathContains implements Validation {
    private static Logger log =
LoggerFactory.getLogger(PathContains.class);

    @Override
    public boolean isValid(HashMap<String, Object> env, Object... params) {
        String path = AutomationUtils.getString(0, params);
        String parentPath = AutomationUtils.getParentPath(env);

        try {
            if (parentPath.startsWith(path)) {
                return true;
            } else {
                return false;
            }
        } catch (Exception e) {
            log.error(e.getMessage(), e);
        }

        return false;
    }
}
```

Create a new action

Any validator should implements interface Action:

```
public interface Action {
    public static final String METHOD_PRE = "executePre";
    public static final String METHOD_POST = "executePost";

    public void executePre(HashMap<String, Object> env, Object... params);
    public void executePost(HashMap<String, Object> env, Object... params);
}
```

- At HashMap<String, Object> env are stored element values map.
- Values set at validator form are stored to String params[].



Refer to AutomationUtils class to get variables. Take a look at the latest doxygen class reference at Developer_Guide.

```
public class ExecuteScripting implements Action {
    private static Logger log =
    LoggerFactory.getLogger(ExecuteScripting.class);




















    @Override
    public void executePre(HashMap<String, Object> env, Object... params) {
    }


    @Override
    public void executePost(HashMap<String, Object> env, Object... params) {
        String script = AutomationUtils.getString(0, params);
        NodeBase node = AutomationUtils.getNode(env);
        String uuid = AutomationUtils.getUuid(env);









        try {
            Interpreter i = new Interpreter();
            i.set("node", node);
            i.set("uuid", uuid);
            i.eval(script);
        } catch (Exception e) {
            log.error(e.getMessage(), e);
        }
    }
}
```


Crontab

With this feature you can create scheduled tasks in a easy way. In this page you can see the already registered tasks. The *last start* and *last stop* columns show the last time a task was executed. If you want to see the output of the script, you can set an email and you will be notified every time the task is executed.



Crontab 

Name	Expression	Type	File Name	Mail	Last Begin	Last End	Active	
Sample 1	* * * * *	BSH	prueba.bsh		29-11-2010 12:07:41 PM	29-11-2010 12:07:41 PM		  
Sample 2	@daily	BSH	prueba2.bsh					  



If you want to test an execution, click on the execute action (flash icon) and this will force the execution. If you have specified an email address, an email also will be sent.



Starting with OpenKM 5.1.2 the field "Type" was removed and will be autodetected from the uploaded file.

When you create or modify a scheduled task, the *last start* and *last stop* entries are reset if you upload a new task definition. Script may be BeanShell (for a small amount of code) or a packed Jar (for more complex logic or bigger scripts). The daemon will look for the **Main-Class** attribute in the Jar's **META-INF/MANIFEST.MF**.

Edit crontab

Name

Sample 1

Mail

Expression

* * * * *

Type

BSH

Active

☐

File

Examinar...

Cancel

Send



Only active tasks will be executed periodically. If you don't want a task to be executed, simple mark it as inactive or remove it.

Expression syntax

Commands are executed by cron when the minute, hour, and month fields match the current time, and when at least one of the two day fields (day of month, or day of week) match the current time. The scheduler examines crontab entries once every minute. The time and date fields are:

Field	Allowed values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names, see below)
day of week	0-7 (0 or 7 is Sun, or use names)

A field may be an asterisk (*), which always stands for *first-last*.

Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive. For example, "8-11" for an *hours* entry specifies execution at hours 8, 9, 10 and 11.

Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: "1,2,5,9", "0-4,8-12".

Step values can be used in conjunction with ranges. Following a range with "/<number>" specifies skips of the number's value through the range. For example, "0-23/2" can be used in the hours field to specify command execution every other hour. Steps are also permitted after an asterisk, so if you want to say "*every two hours*", just use "*/2".

Names can also be used for the *month* and *day of week* fields. Use the first three letters of the particular day or month (case doesn't matter). Ranges or lists of names are not allowed.



The day of a command's execution can be specified by two fields: day of month, and day of week. If both fields are restricted (i.e., aren't *), the command will be run when either field matches the current time. For example, "30 4 1,15 * 5" would cause a command to be run at 4:30 am on the 1st and 15th of each month, plus every Friday.

Instead of the first five fields, one of eight special strings may appear:

String	Meaning
@yearly	Run once a year, "0 0 1 1 *"
@annually	(same as @yearly)
@monthly	Run once a month, "0 0 1 * *"
@weekly	Run once a week, "0 0 * * 0"
@daily	Run once a day, "0 0 * * *"
@midnight	(same as @daily)
@hourly	Run once an hour, "0 * * * *"

BeanShell sample

Create a file called *beanshellSample.bsh* with this content:

```
for (int i=0; i<5; i++) {
    print("Iteration: " + i + "<br/>");
}
```

Now go to Administration > Crontab and create an entry with this input:

- **Name:** BeanShell Sample
- **Mail:** yourmail@domain.com
- **Expression:** */5 * * * *
- **Active:** On

This will create a crontab job which will execute the contents of the *beanshellSample.bsh* file every 5 minutes. The output will be sent to the defined email address.

JAR sample

Create a Java project with a class called *JarSample.java* with this content:

```
package com.openkm.sample;

public class JarSample {

    public static void main(String[] args) {
        System.out.println(cronTask());
    }

    public static String cronTask() {
        StringBuilder sb = new StringBuilder();

        for (int i=0; i<5; i++) {
            sb.append(i).append("<br/>");
        }

        return sb.toString();
    }
}
```



The crontab daemon will look for a static method called **cronTask** with no arguments to be executed. The string returned by this method will be included in the crontab execution result email.



Since OpenKM 6.4 this **cronTask** method has an String argument with the system token.

Now create the JAR. From Eclipse you can use the contextual menu **Export...** and select **Java > JAR file**. Follow the wizard selecting the previously created class (com.openkm.sample.JarSample) as the main class in the MANIFEST.MF.

Go to Administration > Crontab and create an entry with this input:

- **Name:** JAR Sample
- **Mail:** yourmail@domain.com
- **Expression:** */5 * * * *
- **Active:** On

This will create a crontab job which will execute the contents of the *JarSample.jar* file every 5 minutes. The output will be sent to the defined email address.

Users examples

Network importing

The scanner saves scans as pdfs to a shared folder ('/home/scanner') on the server then this script picks them up, loads them into an OpenKM folder ('/okm:root/Scans') and deletes the originals. I am also going to set up a CUPS print to PDF file queue that will print documents into OpenKM using the same method.

Note: This script only loads PDF files. If you want other files loaded then change the `OnlyExt` class to accept other file types

More information at OpenKM Forum ^[1].

```
import javax.jcr.*;
import com.openkm.core.*;
import com.openkm.api.OKMDocument;
import java.io.*;

String token = JcrSessionManager.getInstance().getSystemToken();
Session session = JcrSessionManager.getInstance().get(token);
OKMDocument document = OKMDocument.getInstance();

public class OnlyExt implements FilenameFilter {
    String ext;

    public OnlyExt(String ext) {
        this.ext = "." + ext;
    }

    public boolean accept(File dir, String name) {
        return name.endsWith(ext);
    }
}

File scans = new File("/home/scanner");
Thread.sleep(10000); // Sleep 10 seconds in case files are still being
                    written

try {
    for (File scan : scans.listFiles(new OnlyExt("pdf"))) {

        try {
```

```

        document.createSimple(token, "/okm:root/Scans/" + scan.getName(),
new FileInputStream(scan));
        scan.delete();
    } catch (Exception e) {
        print("Exception: " + e);
    }
}

} catch (Exception e) {
    print("Exception: " + e);
}
}

```

References

[1] <http://forum.openkm.com/viewtopic.php?f=5&t=4555&start=60>

OMR templates

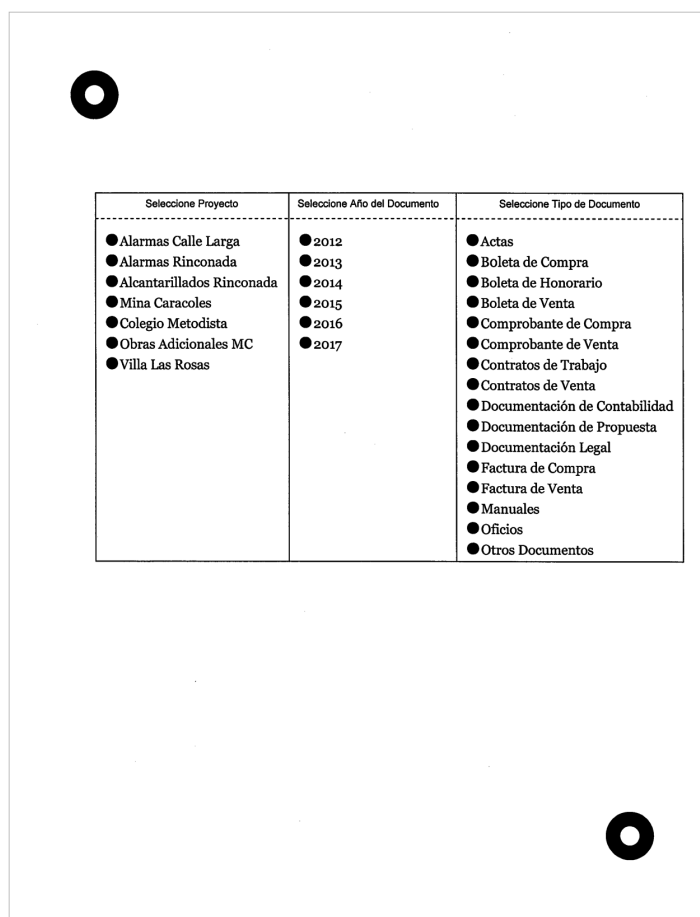
OpenKM integrates OMR (Optical Mark Recognition) that has been written by Aaditeshwar Seth from the Udai Waterloo chapter, at the University of Waterloo. Udai is a volunteer student organization at the University of Waterloo in Canada and the University of California at San Diego. For more information about it see OMRProj website ^[1].

OMR (Optical Mark Recognition) engines is especially suited for extracting data from forms that have been photo-copied and then scanned; hence, the resultant images are likely to suffer from rotations, smudge marks, and random lines here and there. Examples of a few forms that the tool is able to handle with 100% accuracy are shown here.

The image shows two examples of OMR templates. The left template is a form for a survey or census, featuring sections for 'Ration card', 'Voter ID card', 'Mother tongue', 'Home', 'Vehicle', 'Electricity', 'Water', 'Toilet', 'Disposal', 'Garbage', 'Comforts', and 'First name'. The right template is a form for a survey or census, featuring sections for 'Ration card', 'Voter ID card', 'Mother tongue', 'Home', 'Vehicle', 'Electricity', 'Water', 'Toilet', 'Disposal', 'Garbage', 'Comforts', and 'First name'.

Step 1 - Create template

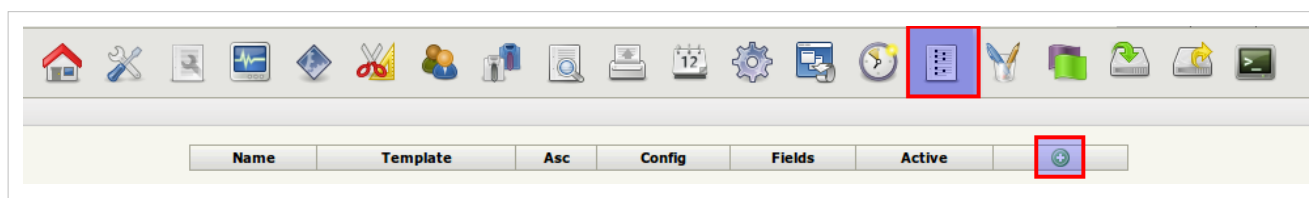
- First make a form in powerpoint using the template . Make sure that you do not move the two concentric circles in the top-left and bottom-right corners of the template. These are used by the software to determine the angle of rotation of the form.
- Take a printout and make as many photocopies of the form as you want.
- On the original printout, color all the marks with a black pen, and scan this marked up printout. Please be as careful as you can to align the printout so that there is no rotation. Some amount of error is fine, but not too much! This scanned image will be the template.



Seleccione Proyecto	Seleccione Año del Documento	Seleccione Tipo de Documento
● Alarmas Calle Larga	● 2012	● Actas
● Alarmas Rinconada	● 2013	● Boleta de Compra
● Alcantarillados Rinconada	● 2014	● Boleta de Honorario
● Mina Caracoles	● 2015	● Boleta de Venta
● Colegio Metodista	● 2016	● Comprobante de Compra
● Obras Adicionales MC	● 2017	● Comprobante de Venta
● Villa Las Rosas		● Contratos de Trabajo
		● Contratos de Venta
		● Documentación de Contabilidad
		● Documentación de Propuesta
		● Documentación Legal
		● Factura de Compra
		● Factura de Venta
		● Manuales
		● Oficios
		● Otros Documentos

Step 2 - Upload template

Click add icon 



Omr template

Create omr template

Template name

Test template

Template

/home/jllort/Escritorio/OMF

Examinar...

Property

Select properties

okp:data.year

okp:data.type

okp:data.project

Active

☒

Cancel

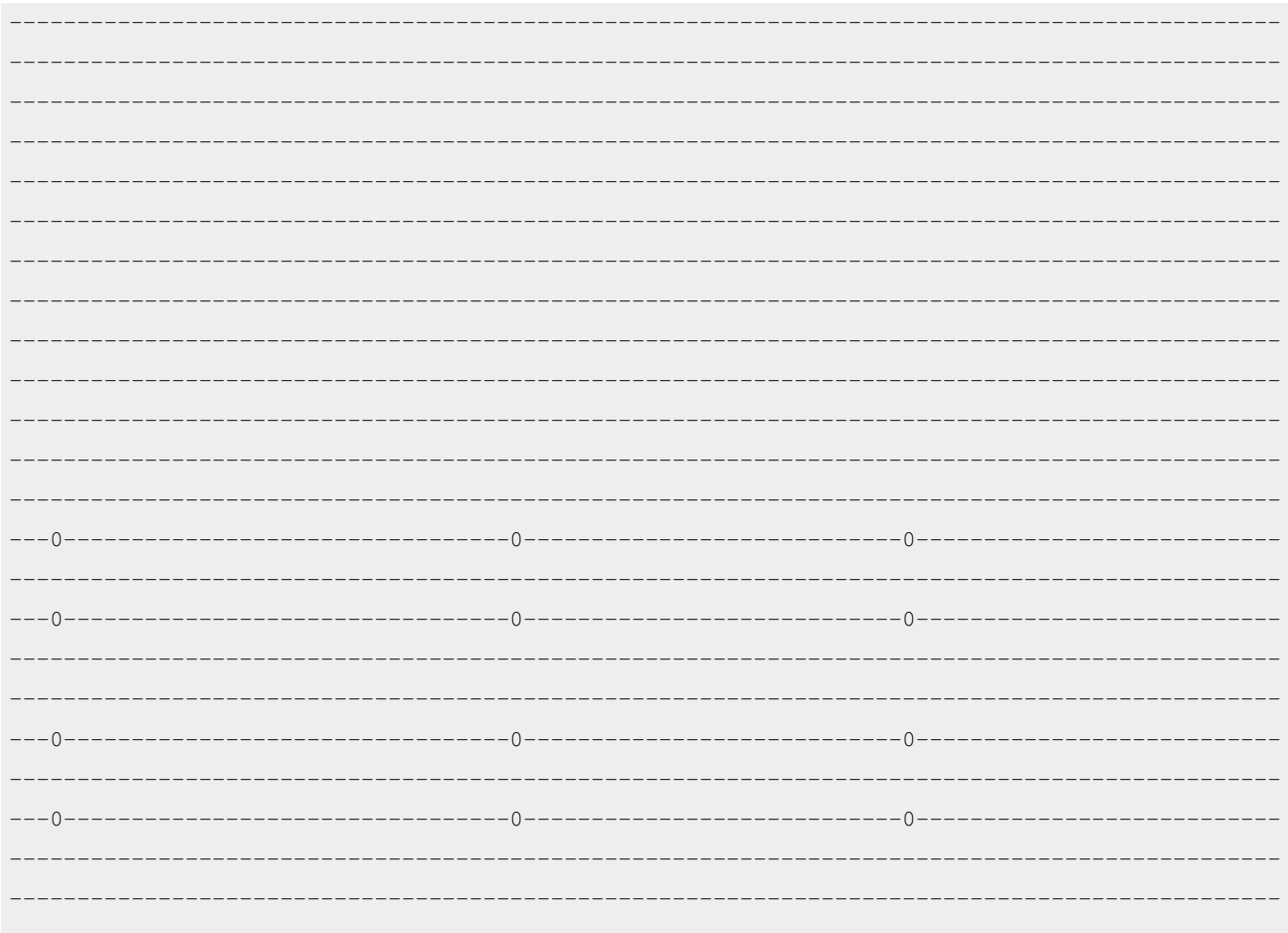
Send

The training process will generate two files .asc and .config

Name	Template	Asc	Config	Fields	Active	
Test template	template.png	template.png.asc	template.png.config		<input checked="" type="checkbox"/>	<div><div></div><div></div><div></div></div>

Step 3 - Update asc file

Download the .asc file and openkm with a standard text editor. This is actually an ascii representation of the template with all the marks marked with a '0', as shown below.




[illegible]




Some of the marks may not be exactly aligned, but don't worry. Now, very carefully, label the marks with single characters, by setting all the marks in the same group with the same character. This is shown below. In case you run out of alphabets, you can use numerals or any other character such as - + ? . etc. Make sure that you do not alter the position of any mark! This is agreeably hard to do, but we chose this method to avoid the need for making a graphical frontend.



[illegible]

Click edit icon  :


Name	Template	Asc	Config	Fields	Active	
Test template	template.png	template.png.asc	template.png.config		✓	


Click edit asc file icon  :

Omr template > Edit omr template

Template name
Template
Property
Asc
Config
Fields
Active

Select properties ▾
okp:data.year ✕
okp:data.type ✕
okp:data.project ✕



 
☒

Select asc file and **click send button**.

Omr template > Edit Asc File

Asc

Step 4 - upload .fields file

You now need to make a [template-filename].fields file, which contains the details of the various form-fields, as follows.

The format of each line is: character subtype name value1 value2....

- Character: This is the same as the character you used in the .asc file
- Type and Subtype: The types of fields supported are single and multiple choice lists, aligned row-wise (horizontally) or column-wise (vertically). And grids aligned horizontally or vertically. Each field is denoted by (i) row single or row multiple for a single or multiple selection field aligned row-wise, (ii) column single or column multiple for a single or multiple selection field aligned column-wise, or (iii) grid row or grid column for a grid field aligned row-wise or column-wise respectively.
- Field name: This can be any name you want to associate with the field.
- Values: These are the field values, in order.



The values has a direct corresponde with metadata values (Property groups), first indicate the property name and others the values into

```
a column single okp:data.project alarmasCalleLarga alarmasRinconada
alcantarilladosRinconada minaCaracoles colegioMetodista
obrasAdicionalesMc villaLasRosas
b column single okp:data.year 2012 2013 2014 2015 2016 2017
c column single okp:data.type actas boletaCompra boletaHonorario
boletaVenta comprobanteCompra comprobanteVenta contratosTrabajo
contratosVenta documentacionContabilidad documentacionPropuesta
documentacionLegal facturaCompra facturaVenta manuales oficios
otrosDocumentos
```

Click edit icon :

Name	Template	Asc	Config	Fields	Active	
Test template	template.png	template.png.asc	template.png.config		✓	

Click edit fields icon :

Omr template
Edit omr template

Template name

Template

Property

Select properties

okp:data.year
✕

okp:data.type
✕

okp:data.project
✕

Asc

Config

Fields

Active
☒



Select fields file and **click send button**.

Omr template
Edit Fields File

Fields

Step 5 - check template

Click check icon  :

Name	Template	Asc	Config	Fields	Active	
Test template	template.png	template.png.asc	template.png.config	template.png.fields		

Upload some file to recognise values:

Omr template

Process Template Check

Upload Form

Examinar...

Cancel

Send

Seleccione Proyecto

☐ Alarmas Calle Larga
☐ Alarmas Rinconada
☐ Alcantarillados Rinconada
☒ Mina Caracoles
☐ Colegio Metodista
☐ Obras Adicionales MC
☐ Villa Las Rosas

Seleccione Año del Documento

☐ 2012
☐ 2013
☒ 2014
☐ 2015
☐ 2016
☐ 2017

Seleccione Tipo de Documento

☐ Actas
☐ Boleta de Compra
☐ Boleta de Honorario
☐ Boleta de Venta
☐ Comprobante de Compra
☐ Comprobante de Venta
☐ Contratos de Trabajo
☐ Contratos de Venta
☐ Documentación de Contabilidad
☒ Documentación de Propuesta
☐ Documentación Legal
☐ Factura de Compra
☐ Factura de Venta
☐ Manuales
☐ Oficios
☐ Otros Documentos

And you'll get the results:

Omr template
Process Template Check

Upload Form
Examinar...

Cancel
Send

RESULTS	
Key	Value
okp:data.year	2014
okp:data.type	documentacionPropuesta
okp:data.project	minaCaracoles


Full example

Can download full example

References


[1] <http://www.cse.iitd.ernet.in/~aseth/udai/OMRProj/README.html>

OCR templates



Since OpenKM 6.4.2 the plug-in system of OpenKM allows you to expand quickly the functionality offered by the platform, extending the available OCR field parsers without having to rebuild the system to add/change the existing functionality. Refer to Extend OCR field parsers if you need to extend OCR field parsers feature on 6.4.2.





















OCR Templates allows to create zonal OCR templates which allows to recognise and extract estructured text from scanned images.





Images should be scanned at least at 200 dpi of resolution to get good text recognition from OCR engine.

Template creation

Open OCR template administration option.

OCR Templates

Name	Active	
		 

Then click on the  new ocr template icon

Name	<input type="text" value="Telefonica invoice"/>		
Image	<input type="text" value="/home/jllort/Escritorio/invo"/>	<input type="button" value="Examinar..."/>	
Active	<input checked="" type="checkbox"/>		
		<input type="button" value="Cancel"/>	<input type="button" value="Create"/>


Fill the form and click **create** button.

Name	Active	
telefonica invoice	✓	✎ - ✓

Add field zones

Then click on the **fields icon**

Template telefonica invoice 🔍




Name	Type	Property	Pattern	
				+ - ✓


Fill the form and select the zone and click **create** button.

Test OCR template

Click on the **check icon** and will be shown all the data fields extracted by zone.

Factura: TA1CV0931947 Fecha: 19 Ene. 06 Agrupación facturable: 300768202																	
Factura: TA1CV0931947 Fecha: 19 Ene. 06 Agrupación facturable: 300768202		<table border="1"> <thead> <tr> <th>Field</th> <th>Type</th> <th>Zone</th> <th>Pattern</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>Invoice numberString</td> <td>TA1CV0931947</td> <td></td> <td></td> <td>TA1CV0931947</td> </tr> </tbody> </table>		Field	Type	Zone	Pattern	Result	Invoice numberString	TA1CV0931947			TA1CV0931947				
Field	Type	Zone	Pattern	Result													
Invoice numberString	TA1CV0931947			TA1CV0931947													
Titular CIF / NIF: 00A61456844 Calle Conxita Supervia, 9 (934907494 INS.ALLI) Bajo Barcelona Teléfono 934092063 Domiciliación bancaria B. SIBIAO VICAYA ARGENTARIA CAPITAN ARENAS, 48 Barcelona Nº de cuenta/libreta 01001150**** Para ser pagado a partir de 19 Ene. 06 *** Oculitos para su seguridad		0810022300768202T-0801-01-008-027788 (934907494 INS.ALLI) Bajo 08028 Barcelona															
Servicios Facturados		Página 1/24															
Red Digital de Servicios Integrados Cuotas Consumos		<table border="1"> <thead> <tr> <th>Importe</th> <th>Sumas</th> </tr> </thead> <tbody> <tr> <td>139,3017</td> <td></td> </tr> <tr> <td>189,4423</td> <td></td> </tr> <tr> <td></td> <td>328,7440</td> </tr> <tr> <td>Total (base imponible)</td> <td>328,7440</td> </tr> <tr> <td>IVA 16%</td> <td>52,5990</td> </tr> <tr> <td>Total a pagar (euros)</td> <td>381,34</td> </tr> </tbody> </table>		Importe	Sumas	139,3017		189,4423			328,7440	Total (base imponible)	328,7440	IVA 16%	52,5990	Total a pagar (euros)	381,34
Importe	Sumas																
139,3017																	
189,4423																	
	328,7440																
Total (base imponible)	328,7440																
IVA 16%	52,5990																
Total a pagar (euros)	381,34																
(El pago de esta factura se acredita por su abono bancario o el recibo de caja)																	

Recognise testing

From main OCR template list, click  **recognise icon**.

Fill the form selecting some scanned image to test recognition.

Upload a file to test recognise level

Image

Examinar...

Level

Slow

✖

Cancel

✔

Recognise

Click **recognise** button.

Name	Confidence level	Control	Recognised
telefonica invoice	100 %	(3 of 3)	✔
travelplan invoice	33 %	(1 of 3)	✖

Enable Zonal OCR data capture

OCR data capture can be enabled from profiles and automation.

Profiles

To enable zonal OCR data capture from profiles should be enabled the check OCR data capture.

The screenshot shows a configuration window with tabs: General, Components, Menu, and Other. The 'General' tab is active. It contains several sections:

- General:** Name (Default), Active (checked).
- Misc:** Quota limit (MB) (0), Advanced filters (checked), Web skin (Default), Print preview (checked), Keywords enabled (checked), Upload notify users (checked), Notify external users (checked), Add documents title (checked), Extensions (Cryptography, Dashboard messaging, Dicom preview, Digital Signature), Reports, Workflows (simple).
- Chat:** Enabled (checked), Auto login (checked).
- Wizard:** Property groups (Digitalización certificada, Invoice, Periodo, Ejercicio), Workflows (simple), Keywords (unchecked), Categories (unchecked), **OCR data capture (checked)** (highlighted with a red box).
- Pagination:** Enabled (checked), Page list (10;25;50;100), Type filter (checked), Misc filter (checked), Show folders by default (checked), Show documents by default (checked), Show mails by default (unchecked), Show records by default (unchecked).

 At the bottom right are 'Cancel' and 'Edit' buttons.



Automation

Automation is divided in two operations, validations and actions.

There's a validation called **IsOCRDataCaptureFile** which validates if OCR data capture engine supports the image format.

Add validation: 						
VALIDATIONS						
Order	Type	active	Param0	Param1	Param2	
0	IsOCRDataCaptureFile					

There're two actions **OCRDataCapture** and **AddOCRDataCaptureToWizard**. **OCRDataCapture** capture data and store to metadata. **AddOCRDataCaptureToWizard** enables end user wizard to see live ocr data capture process.

Add action: AddOCRDataCaptureToWizard 						
ACTIONS						
Order	Type	active	Param0	Param1	Param2	
0	OCRDataCapture		:	:	:	 

For more information take a look at Automation.

Extend OCR field parsers



From OpenKM 6.4+ the Plug-in system of OpenKM allows you to quickly expand the functionality offered by the platform, extending the available OCR field parsers without having to rebuild the system to add/change the existing functionality.

Step 1 - Create class

- The new class should be under `com.openkm.ocr.template.parser` (this is mandatory)
- **Important** do not forget to set the tag `@PluginImplementation` before the class definition

```
package com.openkm.ocr.template.parser;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import net.xeoh.plugins.base.annotations.PluginImplementation;

import com.openkm.dao.bean.OCRTemplateField;
import com.openkm.ocr.template.OCRParserEmptyValueException;
import com.openkm.ocr.template.OCRTemplateException;
import com.openkm.ocr.template.OCRTemplateParser;

/**
 * StringParser
 *
 */
@PluginImplementation
public class StringParser implements OCRTemplateParser {

    @Override
    public Object parse(OCRTemplateField otf, String text) throws
OCRTemplateException, OCRParserEmptyValueException {
        if (text == null || text.equals("")) {
            throw new OCRParserEmptyValueException("Empty
value");
        }

        if (otf.getPattern() == null ||
```

```
otf.getPattern().equals("")) {
    return text != null ? text.trim() : null;
} else {
    Pattern pattern = Pattern.compile(otf.getPattern(),
Pattern.UNICODE_CASE);
    Matcher matcher = pattern.matcher(text);

    if (matcher.matches()) {
        return text != null ? text.trim() : null;
    } else {
        throw new OCRTemplateException("Bad format,
parse exception");
    }
}

@Override
public String getName() {
    return "String";
}

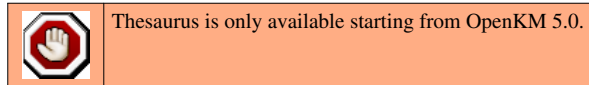
@Override
public boolean isPatternRequired() {
    return false;
}
}
```

Step 2 - Publish

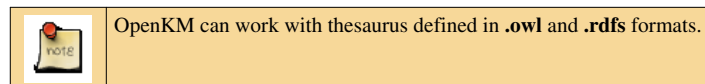
Create your own jar and copy into \$TOMCAT_HOME/lib folder.

Thesaurus

A **thesaurus** is a book that lists **words grouped together according to similarity of meaning** (containing synonyms and sometimes antonyms), in contrast to a dictionary, which contains definitions and pronunciations.



In Information Science, Library Science, and **Information Technology**, **specialized thesauri are designed for information retrieval**. They are a type of controlled vocabulary, for indexing or tagging purposes. If you want more information about the term you can take a look at <http://en.wikipedia.org/wiki/Thesaurus>.



There's a great utility <http://protege.stanford.edu/> for creating and maintaining your own thesaurus. Specific thesauri can be found on the Internet. For example, Agrovoc^[1] (FAO), NASA Thesaurus^[2], UNESCO Thesaurus^[3], Public administration Thesaurus^[4] among others. Some of them are free and some have usage restrictions.

In order to correctly configure OpenKM to use a thesaurus the following configuration properties must be set:

```
kea.thesaurus.owl.file
kea.thesaurus.base.url
kea.thesaurus.tree.root
kea.thesaurus.tree.childs
```

Set OWL file

```
kea.thesaurus.owl.file=vocabulary/file.owl
```

Set base url

```
kea.thesaurus.base.url=http://www.someweb.org
```

Query to get root node

```
kea.thesaurus.tree.root=SELECT DISTINCT UID, TEXT FROM (UID) Y
{OBJECT}, {UID} rdfs:label {TEXT} ; [rdfs:subClassOf {CLAZZ}] where not
bound(CLAZZ)
and lang(TEXT)="en" USING NAMESPACE foaf=<http://xmlns.com/foaf/0.1/>, dcterms=<http://purl.org/dc/terms/>,
rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>, owl=<http://www.w3.org/2002/07/owl#>, rdfs=<http://www.w3.org/2000/01/rdf-schema#>,
skos=<http://www.w3.org/2004/02/skos/core#>, dc=<http://purl.org/dc/elements/1.1/>
```

Query to get childs node

```
kea.thesaurus.tree.childs=SELECT DISTINCT UID, TEXT FROM (UID)
rdfs:subClassOf {CLAZZ}, {UID} rdfs:label {TEXT} where
xsd:string(CLAZZ) =
"RDFparentID" and lang(TEXT)="en" USING NAMESPACE foaf=<http://xmlns.com/foaf/0.1/>, dcterms=<http://purl.org/dc/terms/>,
rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>, owl=<http://www.w3.org/2002/07/owl#>, rdfs=<http://www.w3.org/2000/01/rdf-schema#>,
skos=<http://www.w3.org/2004/02/skos/core#>, dc=<http://purl.org/dc/elements/1.1/>
```

You could be interested in:

- A Method for Converting Thesauri to RDF/OWL^[5]

- OWL to SKOS online conversor ^[6]
- SKOS to OWL online conversor ^[7]
- SKOS Simple Knowledge Organization System ^[8]

Also check Thesaurus full example 📁 for a step-by-step guide.

References

- [1] <http://aims.fao.org/website/AGROVOC-Thesaurus/sub>
- [2] <http://www.sti.nasa.gov/thesfrm1.htm>
- [3] <http://www2.ulcc.ac.uk/unesco/>
- [4] <http://redined.r020.com.ar/es/index.php?tema=55>
- [5] <http://thesauri.cs.vu.nl/>
- [6] <http://owl.cs.manchester.ac.uk/owltoskos/>
- [7] <http://www.ebusiness-unibw.org/tools/skos2owl/>
- [8] <http://www.w3.org/2004/02/skos/>

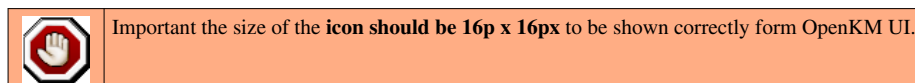
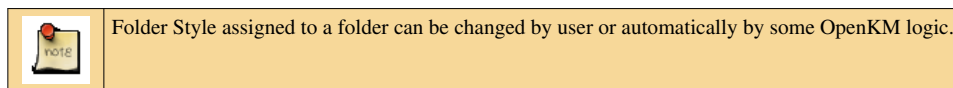
Folder style admin

Folder Style allows to change folder style icon.

Introduction

OpenKM take in consideration Folder style icon in concordance of folder status. For each folder status can be set a distinct icon.

There's a special Folder Style called "Default" which can not be deleted or renamed. This Folder Style is used by default for all folders in UI which does not have assigned any Folder Style.

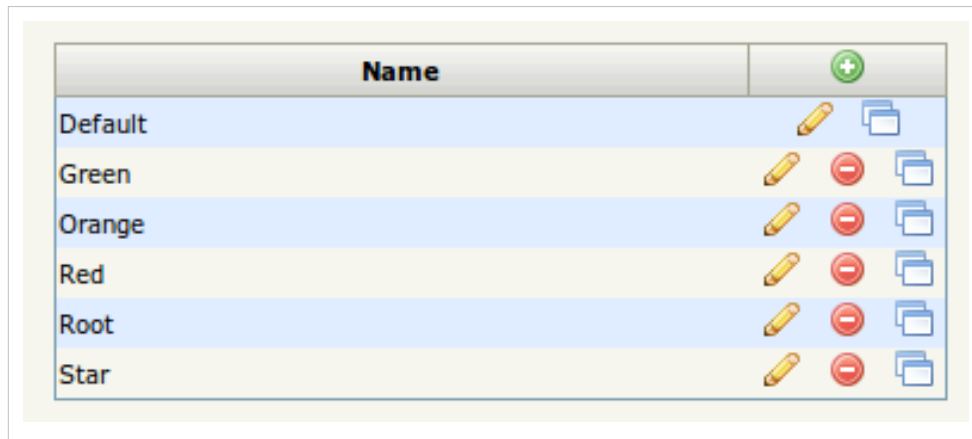
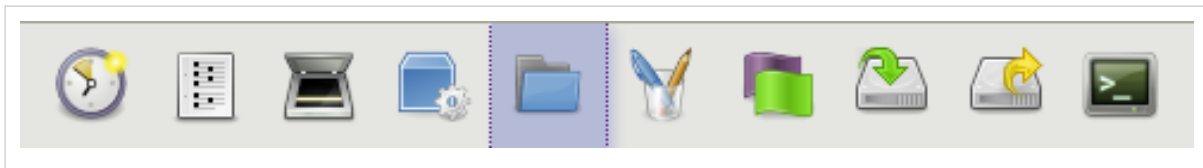


Available folder status:

- Read-only with children icon
- Read-only with subscription icon
- Read-only with children and subscription icon
- Read-write icon
- Read-write with children icon
- Read-write with subscription icon
- Read-write with children and subscription icon

Create

Folder Style administration option.




Then click on the new folder style icon

Fill the form and click **create** button.

Name	Tag Green
Read-only icon	
Read-only with children icon	
Read-only with subscription icon	
Read-only with children and subscription icon	
Read-write icon	
Read-write with children icon	
Read-write with subscription icon	
Read-write with children and subscription icon	



A new folder style has been created with empty icons. Meanwhile all icons will not be set, the new folder style will not be available from UI.

Then click on the  **create icon** until all icons will be set.

Folder style: Tag Green

Read-only icon


Copy to all icons ☐

Image

Examinar...

Cancel

Create



The checkbox **Copy to all icons** can be used to set this icon for all icon status

Name

Read-only icon

Read-only with children icon

Read-only with subscription icon

Read-only with children and subscription icon

Read-write icon

Read-write with children icon

Read-write with subscription icon

Read-write with children and subscription icon

Cancel

Edit

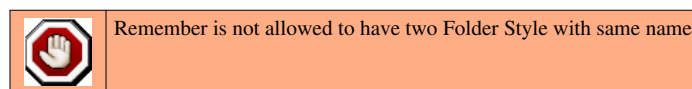
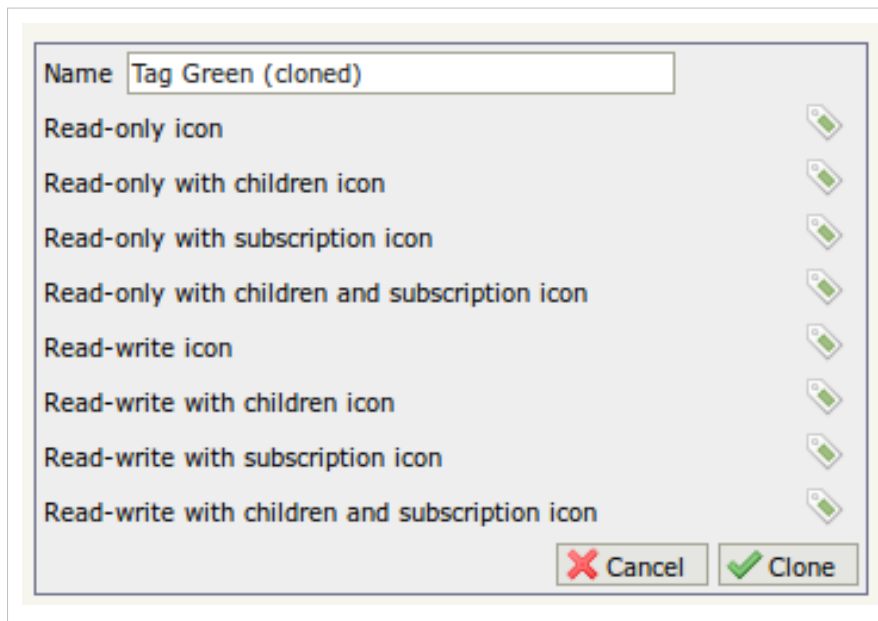
Done

Finally click on **done** button.

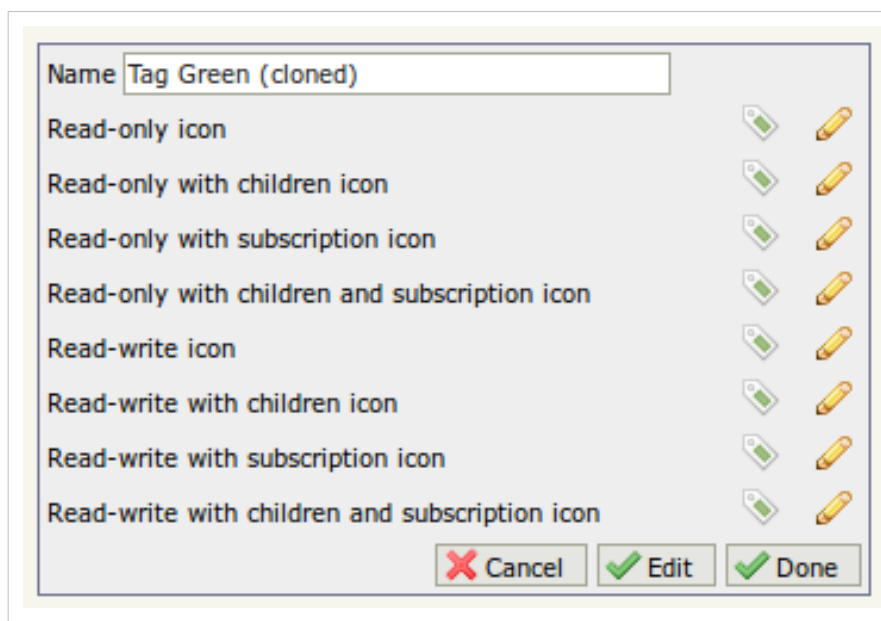
Name			
Default			
Green			
Orange			
Red			
Root			
Star			
Tag Green			

Clone

Click on the  **clone icon**



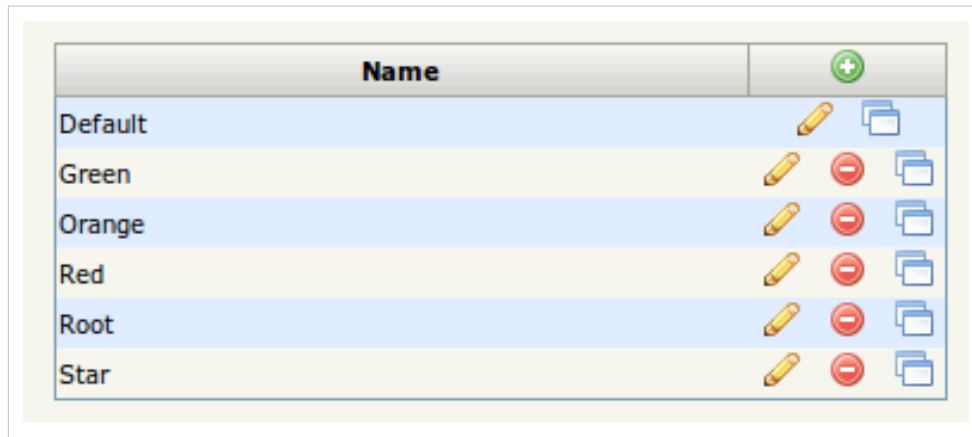
Click on **clone** button.



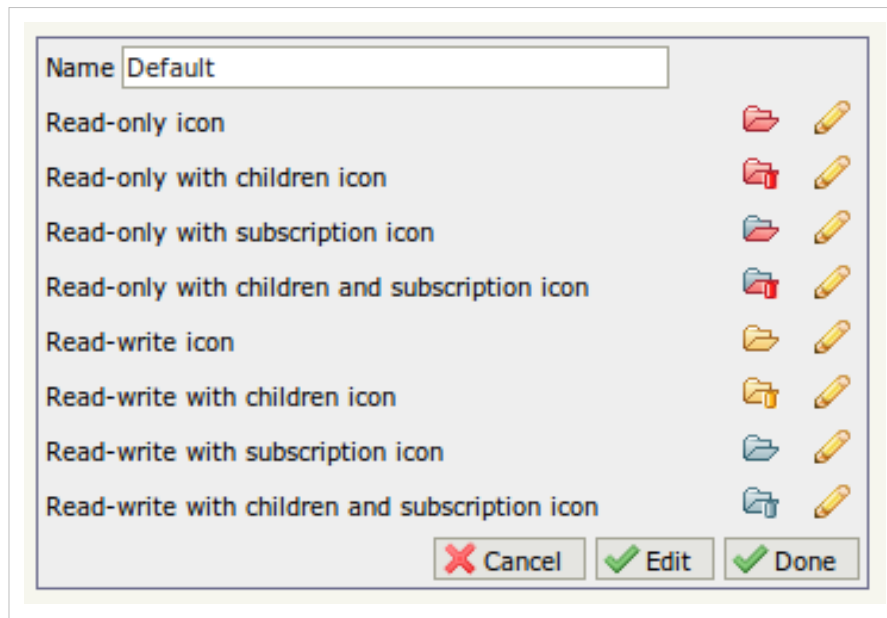
Finally click on **done** button.

Edit

Click on the  **edit icon**



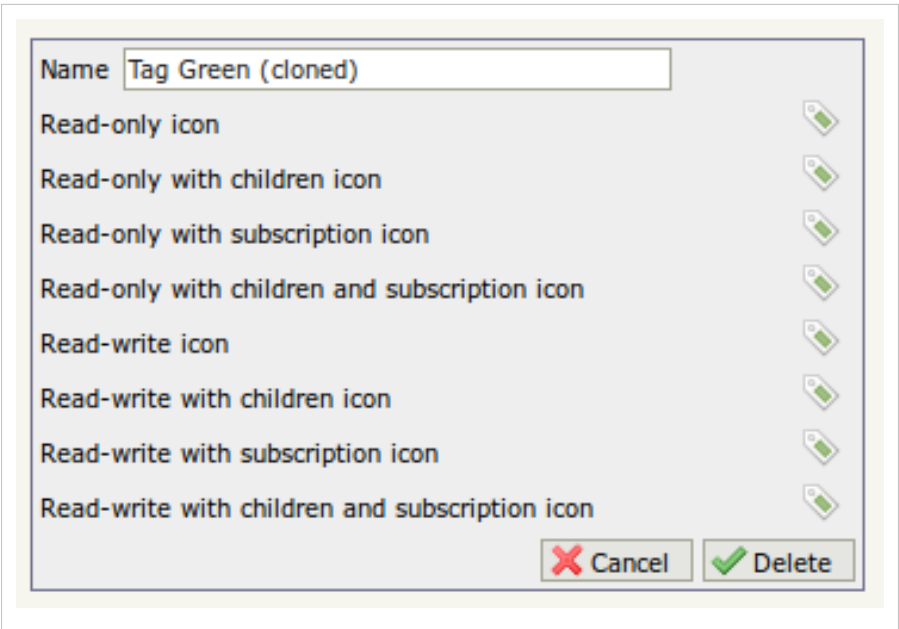
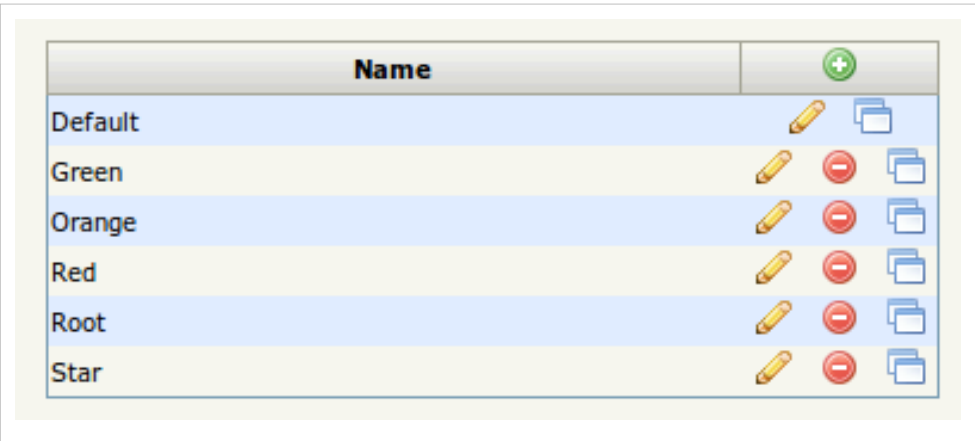
Change all icons as you wish.



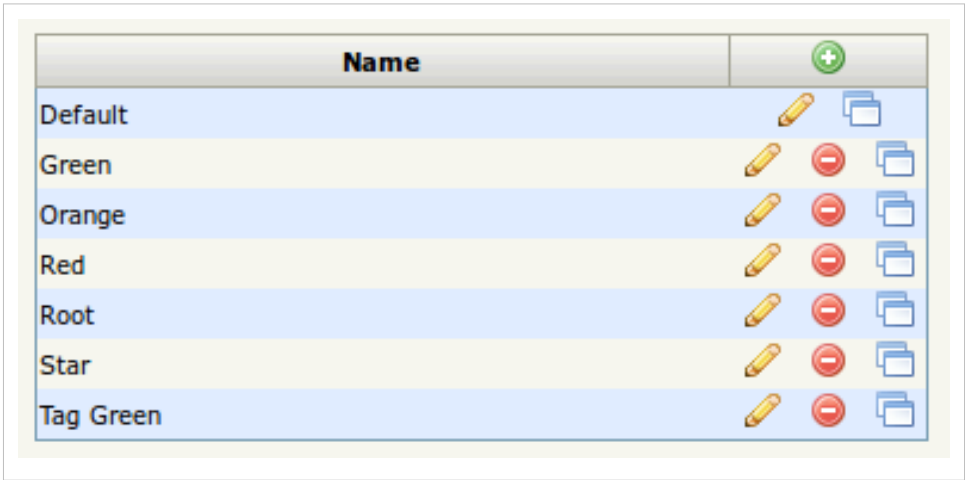
Finally click on **done** button.

Delete

Click on the  delete icon



Finally click on **delete** button.



Generate thesaurus

Thesaurus must be properly configured in the OpenKM.cfg file. For more information on how to do it, refer to the Thesaurus section. By default thesaurus folder structure is not created. It must be created in administration view. In order to create it, these properties must have correct values:

```
kea.thesaurus.owl.file
kea.thesaurus.base.url
kea.thesaurus.tree.root
kea.thesaurus.tree.childs
```

If these properties are well defined in OpenKM.cfg, then the administrator should execute "Send" to start thesaurus folder tree creation. The level indicates the tree depth. A log will be displayed on the screen to monitor thesaurus folder creation. This process could take some hours depending on your thesaurus size.

Generate thesaurus

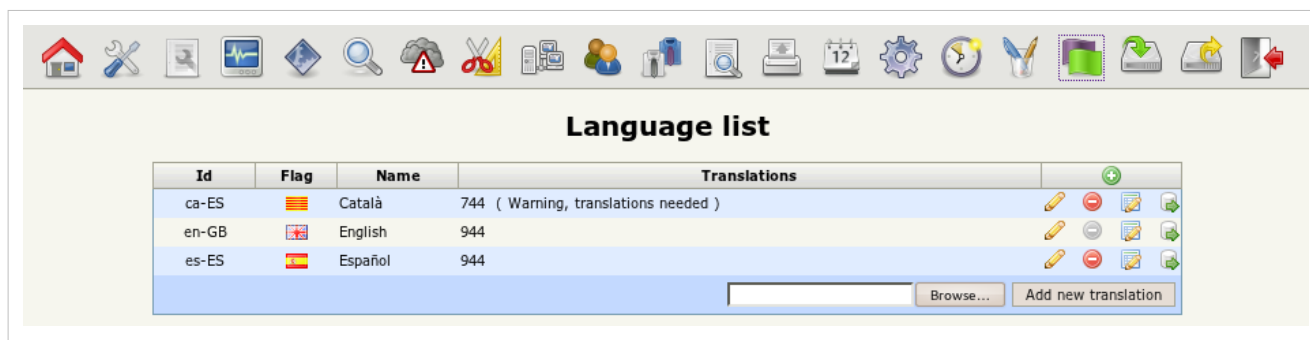
Show level 1

Send

Parameter	Value
kea.thesaurus.skos.file	/vocabulary/ag_skos_20070219.rdf
kea.thesaurus.owl.file	/vocabulary/agrovoc_oei2007.owl
kea.thesaurus.vocabulary.serql	SELECT X,UID FROM {X} skos:prefLabel {UID} WHERE lang(UID) = "en" USING NAMESPACE rdf=, skos=, rdfs=, dc=, dcterms=, foaf=
kea.thesaurus.base.url	http://www.fao.org/aos/agrovoc
kea.thesaurus.tree.root	SELECT DISTINCT UID, TEXT FROM {UID} Y {OBJECT}, {UID} rdfs:label {TEXT} ; [rdfs:subClassOf {CLAZZ}] where not bound(CLAZZ) and lang(TEXT)="en" USING NAMESPACE foaf=, dcterms=, rdf=, owl=, rdfs=, skos=, dc=
kea.thesaurus.tree.childs	SELECT DISTINCT UID, TEXT FROM {UID} rdfs:subClassOf {CLAZZ}, {UID} rdfs:label {TEXT} where xsd:string(CLAZZ) = "RDFparentID" and lang(TEXT)="en" USING NAMESPACE foaf=, dcterms=, rdf=, owl=, rdfs=, skos=, dc=
kea.model.file	/vocabulary/ag_skos_20070219.model
kea.automatic.keyword.extraction.number	10
kea.automatic.keyword.extraction.restriction	on
kea.stopwords.file	/vocabulary/stopwords_en.txt

Language

Adding new language translations, creating your own or simply modifying what comes by default is so easy with OpenKM.



Add new language

Then click on the new language icon

Create language

Id

Name

Flag

And simply fill the data with your international language code. If you don't know which is yours, take a look here <http://www.i18nguy.com/unicode/language-identifiers.html> to find your language code.

Image icon should be 16x11 pixels take a look on <http://www.famfamfam.com/lab/icons/silk/>

Edit a language

Then click on the edit language icon

Edit language

Id

Name

Flag

Delete a language

Then click on the  delete language icon

Delete language

Id

es-ES

Name

Español


Flag



Browse...

Cancel

Send



English language is the base language and can not been deleted

Edit translation

Then click on the  edit translation icon

Translation list				
#	Module	Key property	English	Català
1	frontend	OKM-001001	Repository internal error	Error intern del repositori
2	frontend	OKM-001002	Folder not found	No existeix una carpeta amb aquest nom
3	frontend	OKM-001003	Folder already exists	Ja existeix una carpeta amb aquest nom
4	frontend	OKM-001006	Repository internal error	Error intern de l'aplicació
5	frontend	OKM-001009	Folder access denied	No teniu permisos per a accedir a la carpeta
6	frontend	OKM-001015	Folder path not found	La ruta de la carpeta no existeix
7	frontend	OKM-001024	Database error	Error en la base de dades
8	frontend	OKM-002001	Repository internal error	Error intern en el repositori
9	frontend	OKM-002002	Document not found	No existeix un document amb aquest nom
10	frontend	OKM-002003	Document already exists	Ja existeix un document amb aquest nom

Import translation

Take a look here Language_Packs. There're some available translations. Simply select one and click on the "add new translation" button.

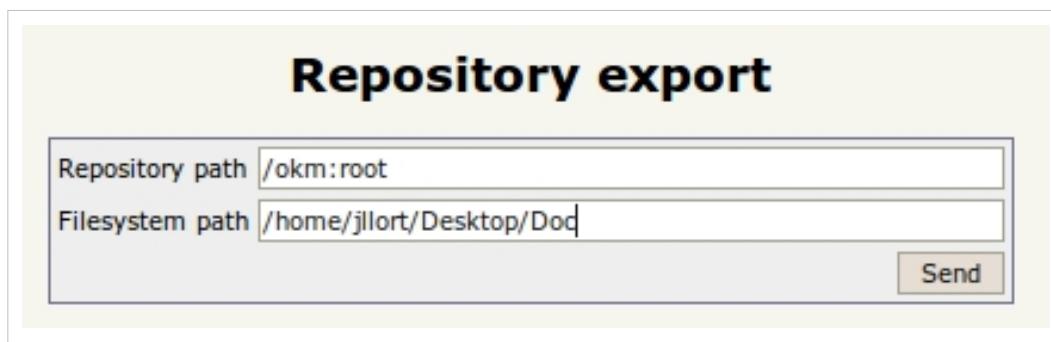
Export translation

Then click on the  export translation icon

You can export the translation file and share it with other users. Take a look here Language_Packs.

Repository export

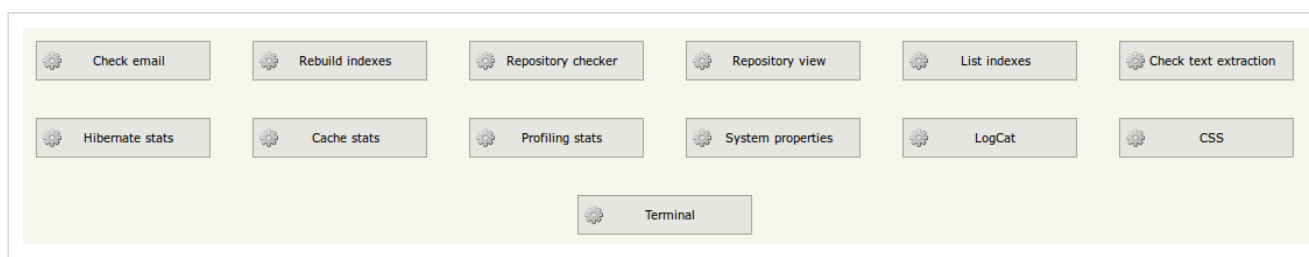
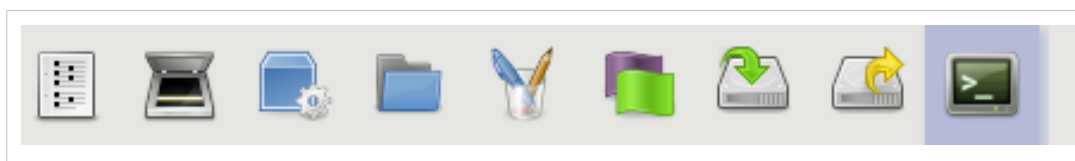
With OpenKM you can export all your folder structures and documents to your server disk. This operation is really easy. You simply put your OpenKM origin path and your operating system destination folder path.



- In the screenshot the server path is based on Linux filesystem. Windows users should use Windows OS filesystem path syntax i.e. c:\some_path\...
- OpenKM import and export features work for the local filesystem, so if you want to export files to another computer you have to copy them from the OpenKM server first.

Utilities admin

From Utilities administration option can access some advanced administrator utilities.

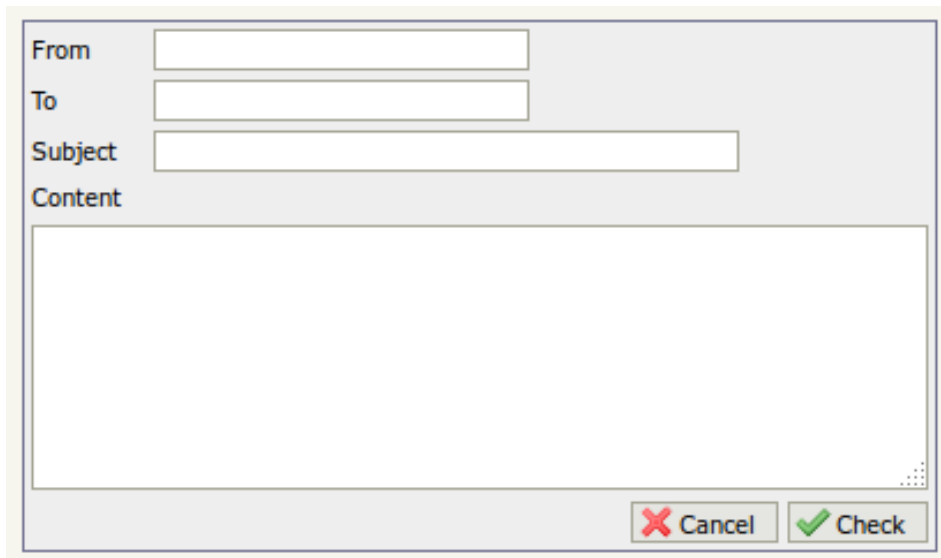


Utilities

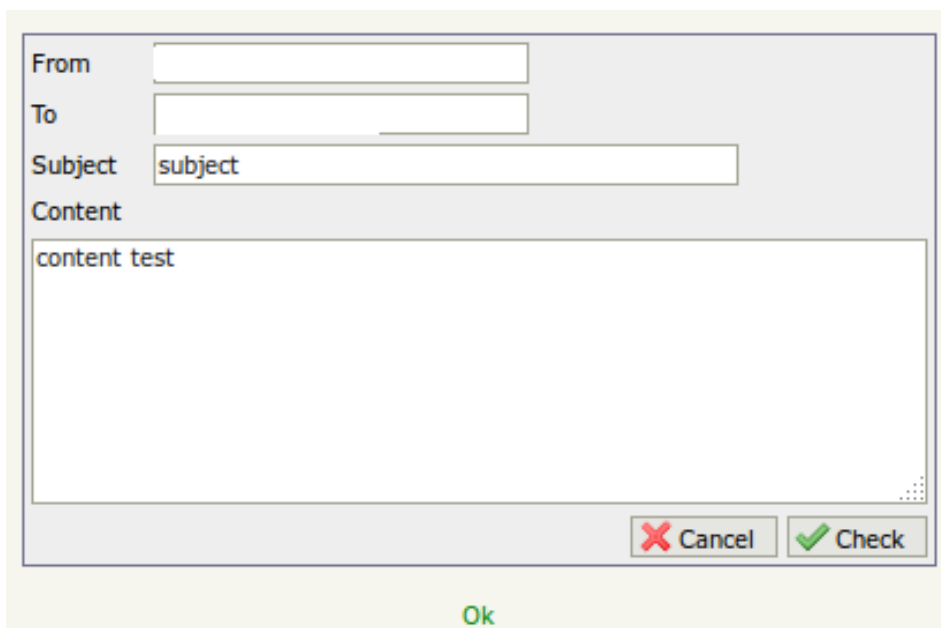
- Check mail To check mail configuration.
- Rebuild indexes Manage lucene indexes.

Utilites admin check mail

Check mail utility is usefull to test mail configuration.



Fill form fields and finally click **check** button.



If all is right check will get back **OK** result.

Application configuration

OpenKM configuration file

OpenKM.cfg file is the main OpenKM configuration file. You can find this file in your server in `$TOMCAT_HOME/OpenKM.cfg`. In older OpenKM releases based on JBoss, the configuration file is located at `$JBOSS_HOME/OpenKM.cfg`.

A full listing of available configuration options can be found when logged into OpenKM with an administrator role. You will only see the **Administration** tab if the logged in user has the **ROLE_ADMIN** (OpenKM 6.2) or **AdminRole** (OpenKM 5.1) role.



Each time you make a change to the OpenKM.cfg file, the application server must be restarted for the change to take effect.



Starting with OpenKM 5.1, these configuration properties are located in the database to ease their management. To change the configuration properties, go to **Administration > Configuration**.

Changing max file upload size

The default is 64MB. If you want to set the limit to 100MB:

```
max.file.size=100.
```

Since OpenKM relies on the `InputStream.available()` method which returns an Integer, and according to Java Primitive Data Types ^[1], you should not upload files bigger than 2GB.

Since OpenKM 5.1.10 you can disable file size checking by setting **max.file.size** to 0.

Since OpenKM 6.0 by default this option is set to 0, so file size check is disabled by default.

Since OpenKM Professional 6.2.21 you can use suffixes to express this amount of bytes. For example, 128MB or 512KB.

Changing repository.xml configuration filename and path

```
repository.config=repository.xml
```

Changing default repository home

OpenKM 6.2

By default, the OpenKM document repository is located at `$TOMCAT_HOME/repository`. To change the repository folder to another location, use the **repository.home** configuration property in the **OpenKM.cfg** file, e.g.

```
repository.home=C:\okmrepo
```

If you want to fine tune the contents of the document repository you can also use these configuration properties in the **OpenKM.cfg** file, e.g

```
hibernate.search.index.home=C:\okmrepo\index
repository.datastore.home=D:\okm\datastore
repository.cache.home=D:\okm\cache
```

OpenKM 5.1 and before

By default the repository is located in the `$JBOSS_HOME/repository` folder but you can change it with:

```
repository.home=repotest
```

Changing default user role

By default, the role required to connect to OpenKM is called **ROLE_USER**. You can change this to another role name. Go to **Administration > Configuration** and set:

```
default.user.role=ROLE_OTHER
```

Changing default admin role

By default, the admin connection role is called **ROLE_ADMIN**. Go to **Administration > Configuration** and set:

```
default.admin.role=ROLE_OTHER_ADMIN
```

You can change this to another role name. This change also needs some changes in the file located at `$TOMCAT_HOME/webapps/OpenKM/WEB-INF/applicationContext.xml`:

```
<security:intercept-url pattern="/admin/**" access="ROLE_ADMIN" />
```

Change max results in UI

This option limits the search results in the UI.

```
max.search.results=25
```

Since: OpenKM 4.0

Change access URL

By default OpenKM sets the access URL to `http://localhost:8080/OpenKM/index.jsp`", **but obviously you don't want to access your OpenKM installation only from localhost. This URL is used in mail notifications and the copy-to-clipboard feature. To change this default URL use this property:**

```
application.url=http://your-server.com/OpenKM/index.jsp
```

Since: OpenKM 4.0

Change principal adapter

OpenKM can handle user access using the Spring Security framework. OpenKM needs an available method for reading users and roles, so when users are stored in a database (as is the default), the class **DatabasePrincipalAdapter** does this job. If you configure OpenKM to authenticate against an LDAP server, you need to configure another principal adapter like **LdapPrincipalAdapter**. For more information take a look at OpenKM authentication and Active Directory.

```
principal.adapter=com.openkm.core.DatabasePrincipalAdapter
```

When using **DatabasePrincipalAdapter** you can create users and roles from OpenKM Administration. In this case you can use the **principal.identifier.validation** configuration property to defined a regular expression to validate user names and roles. By default is set to `^[a-zA-Z0-9_]+$` to prevent inclusion of potential problematic characters.

Since: OpenKM Professional 6.2.13

Switch OpenKM to demo mode

In demo mode some options are disabled by default.

```
system.demo=on
```

Since: OpenKM 4.0



Available only in OpenKM Professional.

Enabling OCR

To enable OCR you must set the path of your OCR engine

```
system.ocr=/usr/local/bin/tesseract
```

Starting with OpenKM 5.1 you can select between 3 OCR engines:

- Tesseract 2.x
- Tesseract 3.x
- Cuneiform 0.7.x

See Third-party software integration: OCR for more info.

Setting default language of OpenKM UI

OpenKM by default guesses the language depending on the browser user language configuration, but you can set a language for all OpenKM users to use by default:

```
default.lang=es-ES
```

Force username to lowercase

You can force all usernames to be written in lower case. It's useful for example when you integrate with microsoft active directory where the username is not case sensitive.

```
system.login.username.lowercase=on
```

Since: OpenKM 4.0

System read only

You can force the whole repository to be read only. This is useful when you're doing maintenance tasks, while doing migration and you want to continue to have the system running but you want to be sure any user will not be able to upload or change documents, etc...

```
system.readonly=true
```

Since: OpenKM 5.0

Enable document preview

Several applications are used by OpenKM to generate the document preview: OpenOffice.org, pdf2swf and convert. Read Enable PDF to SWF conversion, Enable image preview and Enable OpenOffice.org service. Also take a look at Random errors in preview in case of problems.

Enable PDF to SWF conversion

To enable the preview UI tab, OpenKM needs to convert PDF files to SWF. You must also have OpenOffice installed as a service.

```
system.pdf2swf=/usr/bin/pdf2swf
```



Starting from OpenKM 5.1 this configuration property has changed to **system.swftools.pdf2swf**.



Since OpenKM 5.1.9 the pdf2swf command line parameters should be added in the configuration property, for example:

```
system.swftools.pdf2swf=/usr/bin/pdf2swf -T 9 -f ${fileIn} -o ${fileOut}
```

Note: Read Third-party software integration: SWFTools for more complete documentation.

Since: OpenKM 4.1

Enable PS to SWF conversion

To enable postscript document preview, OpenKM needs to convert PS files to SWF using the **ps2pdf** utility from Ghostscript:

```
system.ghostscript.ps2pdf=/usr/bin/ps2pdf
```

Since: OpenKM 5.1.2

Enable image preview

To enable image preview, you need to install the ImageMagick *convert* utility and configure it:

```
system.convert=/usr/bin/convert
```



Starting from OpenKM 5.1 this configuration property has changed to **system.imagemagick.convert**.

Since: OpenKM 4.1

Enable OpenOffice.org integration

OpenOffice.org is used by OpenKM to convert files to PDF and generate the preview.

```
system.openoffice=on
```

Starting with OpenKM 5.0 this property was changed to:

```
system.openoffice.path=/usr/lib/openoffice
system.openoffice.tasks=5
system.openoffice.port=2222
```

Where you should specify an OpenOffice.org installation path, the maximum number of conversion tasks before restarting the service and a port where the OpenOffice.org conversion service will be attached. Only the first property is mandatory.



If you have problems with OpenOffice / LibreOffice integration, take a look at OpenOffice configuration issues.

Note 1: Read Third-party software integration: OpenOffice.org for more up-to-date documentation.

Note 2: You can enhance OCR results by configuring an OpenOffice.org dictionary. See OpenKM 5.1 OCR configuration for more info.

Since: OpenKM 4.0

Configuring email

To configure email service, you must set subscription and notification properties. View a complete example in Notification and subscription messages.

```
subscription.message.subject=OpenKM - {0} - {1};
subscription.message.body=Document: <a href=\"{0}\">{1}</a>
notify.message.subject=OpenKM - NOTIFICATION - {0}
notify.message.body=Document: <a href=\"{0}\">{1}</a>
```

Since: OpenKM 4.1

Configuring password validation

To configure password validation, you have several properties. CompletePasswordValidator is the password validator class that comes by default with OpenKM. By default, this validator is not enabled. You can also create your own Create password validator 🗝️.

You can select which validation rules you want to enable, for example if you only enable validator.password.min.length property, the validator will only validate minimum length.

```
validator.password=com.openkm.validator.password.CompletePasswordValidator
validator.password.min.length=numeric value greater than 0
validator.password.max.length=numeric value greater than 0
validator.password.min.lowercase=numeric value greater than 0
validator.password.min.uppercase=numeric value greater than 0
validator.password.min.digits=numeric value greater than 0
validator.password.min.special=numeric value greater than 0
```

Since: OpenKM 4.1

Configuring the chat service

By default chat and chat autologin are enabled. In order to enable or disable them, values can be "on" or "off". You can change this from Administration > Profile.

Since: OpenKM 6.2

Configuring schedulers

There are some schedulers that OpenKM uses. Some internally, like mail importer and getting repository news, and others used by the user interface like dashboard refresh time and keepalive.

KeepAlive is used by the user interface to maintain the browser connected to OpenKM even when a user stays for some time without performing any operation. For this reason, it should never be lower than server session timeout (by default set to 15 minutes).

```
schedule.session.keepalive=15
```

Dashboard data is refreshed by default every 30 minutes.

```
schedule.dashboard.refresh=30
```

Since: OpenKM 4.1

Other repetitive actions like user mail importing are now managed from Crontab. So you can enable, disable or change the run frequency from Administration.

Since: OpenKM 6.2

Configuring wizard

You can configure a wizard to be used each time a user uploads a document. The wizard can be a sequence of property groups, keywords (including thesaurus) and categories. By default, wizard is disabled. You can enable wizard for properties, keywords and categories from Administration > Profile.

Since: OpenKM 6.2

Customizing application logo

You can replace the default OpenKM logo in the login page and reports.

```
logo.login  
logo.mobile  
logo.report  
logo.tiny
```

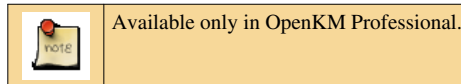
Recommended image size:

- logo.login: 316px × 74px (background #F1F3F5)
- logo.mobile: 161px × 38px (background #F1F3F5)
- logo.report: 150px × 35px (background #FFF000)
- logo.tiny: 51px × 13px (background #E5E5E1)

You can also set a message on the login page with:

```
logo.text
```

Since: OpenKM 5.1 (configuration stored in database)



Configuring WebDAV access

In recent OpenKM releases, WebDAV is disabled by default. If you want to enable it, set the property **system.webdav.server** to on. The *okm:root* path may cause problems with some WebDAV clients. For this reason there is another configuration parameter **system.webdav.fix** which replaces a path like *okm:root* with *okm_root*. See WebDAV access for more info.

Since: OpenKM 5.1

Disable document name mismatch check

By default OpenKM will check the document name on check-in to ensure you upload another version of the same document. You can disable this behaviour with this configuration property:

```
system.document.name.mismatch.check=false
```

Since: OpenKM 5.0.1

Force keywords lowercase conversion

By default keywords are stored as is, but you can force a lowercase conversion using this configuration property:

```
system.keyword.lowercase
```

Since: OpenKM 5.1

Disable user assign on document creation

By default when a user creates a document or folder, he is added to the node with full permissions. You can disable this behaviour this way:

```
user.assign.document.creation=off
```

Since: OpenKM 5.0.2

Improve OpenKM performance

If you experience a slow down of OpenKM when the number of documents grows, you can disable user quotas or make use of the user items size cache.

To enable user items size cache:

```
user.item.cache=on
```

Starting from 5.1 **user.item.cache** is set to **on** by default.

Starting from OpenKM 5.1.6, you can disable user quotas. Go to User Profiles (**Administration -> Profiles**) and set **Quota limit** to 0 (This field is in megabytes).

Since: OpenKM 5.0.3

Change conversion cache home

In case you have little space in your application server partition, you can configure to store the conversion cache folder somewhere else. For this, use the **repository.cache.home** configuration property. By default, this is set to `$TOMCAT_HOME/repository/cache`.

Since: OpenKM 5.1.8

Re-indexing the whole repository

To re-index the whole document repository (document content will be extracted again and indexed by Lucene), follow these steps, depending on your OpenKM version:

OpenKM 6.2

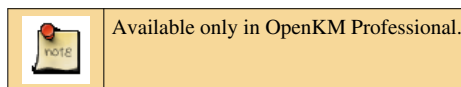
- Log in as administrator.
- Go to Administration > Utilities > Rebuild indexes.
- Select the **Lucene indexes** option and click on **Send**.
- A progress indicator will be shown and the repository will be put in read-only mode until re-indexing is completed.

OpenKM can use two different ways of rebuilding Lucene indexes: sequential or parallel. By default sequential re-indexing is enabled, but you can select the mode using the **hibernate.indexer.mass.indexer** configuration property.

In **sequential mode** or you can use the **hibernate.indexer.batch.size.load.objects** configuration property to indicate to Hibernate how many object should handle every time. To avoid *OutOfMemory* problems, the repository needs to be re-indexed in batch. If the value of this property is too low, the performance will be bad but if is too high you can have *OutOfMemory* problems.

In **parallel mode** you have several configuration properties to tune its performance:

- **hibernate.indexer.batch.size.load.objects**: batch size used to load the root entities. Defaults to 30.
- **hibernate.indexer.threads.subsequent.fetching**: number of threads used to load the lazy collections related to the indexed entities. Defaults to 8.
- **hibernate.indexer.threads.load.objects**: number of threads used to load the root entities. Defaults to 4.
- **hibernate.indexer.threads.index.writer**: number of threads used to analyze the documents and write to the index. Defaults to 3.



Available only in OpenKM Professional.

OpenKM 5.1



Keep in mind that the next time you start JBoss, a process which runs the text extraction for every document in the repository will be launched and you can't use OpenKM until this process is finished.

- Stop JBoss.
- Delete the `$JBoss_HOME/repository/repository/index` and `$JBoss_HOME/repository/workspaces/default/index` directories.
- Start JBoss again.

Security configuration

Starting with OpenKM 6.0 you can make use of these configuration properties:

- **security.access.manager:** You can switch between "simple" and "recursive". This is the way the security is evaluated. With "recursive", the security is evaluated starting with the selected node until the repository root. The "simple" option only evaluates the selected node. By default, this is set to "simple".
- **security.search.evaluation:** Here you can choose between several search result security evaluation strategies. The fastest one is called "lucene", because it is the Lucene search engine which restricts the search results based on the node security. The others are "am_more", "am_window" and "am_limited", which make use of the AccessManager (see previous property) to restrict the search results. By default, this is set to "lucene".

By default OpenKM 6 is configured this way:

```
security.access.manager = simple
security.search.evaluation = lucene
```

Which provides great performance by only evaluating a single node. But if you want to evaluate every parent you can change this behavior with these settings:

```
security.access.manager = recursive
security.search.evaluation = [ am_more | am_window | am_limited ]
```











This will reduce OpenKM performance (at least in search) but will match customer requirements. Performance can be improved customizing **com.openkm.cache.parentNodePermissions** cache region definition.

Also provided is a new AccessManager: **read_recursive** used to prevent showing documents in search results when any folder in the path doesn't have read access for the user who ran the query. Should be used in combination with **am_more**.

```
security.access.manager = read_recursive
security.search.evaluation = am_more
```

Difference between "simple" and "recursive"

Given this repository structure:

-  okm:root
 -  level-1a
 -  level-2a
 -  level-3a
 -  alpha.doc
 -  beta.doc
 -  gamma.doc
 -  level-3b
 -  level-1b
 -  level-1c

When using *simple* if you revoke permission from the *level-2a* folder and the user tries to view the *level-3a* folder, OpenKM will still show the documents to this user. When the user performs a search, OpenKM will return these documents as search results. However, the user can't access the documents because when OpenKM tries to access the document, it will open every path location and will fail to open *level-2a* because the user has no rights. You need to apply revoke recursively to prevent the user from seeing these documents, so they won't be shown in the search results.

When using *recursive* if you revoke permission from the *level-2a* folder and the user tries to view the *level-3a* folder, OpenKM will report an error because Access Manager also evaluates the permission of the *level-2a* ancestor node. When the user performs a search, OpenKM won't show these documents in the search results. Keep in mind that this Access Manager implementation is slower because it needs to evaluate more nodes: the more node depth the more parent nodes need to be evaluated.

Since: OpenKM 6.2

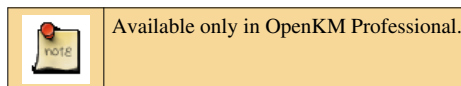
Extended security

Starting with OpenKM 6.0 you can extend security to download files, start workflows, add, remove or modify property groups (metadata) or compact history.

The possible values are:

- DOWNLOAD = 1024;
- START_WORKFLOW = 2048;
- COMPACT_HISTORY = 4096;
- PROPERTY_GROUP = 8192;
- **security.extended.mask:** Here you can activate which kind of extended security you want in your OpenKM. For example to enable download and property groups you should put the mask 9216 (1024+8192)

Since: OpenKM 6.0.1



Managing security

By default when you change the security of a node (document or folder) in OpenKM, every time you click to grant or revoke a permission, the action is performed by OpenKM. This means that if you want to change the security of 10 users, 10 commands are sent to the server to be performed. The case is worse when every change affects several hundred (or even thousands) of nodes.

In OpenKM Professional 6.2.9 you can configure an alternative security mode where the changes are only applied at the end, so it is faster. To enable this new mode you need to use the **security.mode.multiple** configuration property. Also, we have included an improvement in OpenKM Professional which can send these security changes as a background task if this takes too long to complete. You can fine tune this behaviour using the **security.live.change.node.limit** configuration property.

Since OpenKM Professional 6.2.11 you can check the progress of these pending task from **Administration > Statistics > Pending tasks queue**.

Since: OpenKM Professional 6.2.9 and 6.2.3 Community

AutoCAD previewer

If you have problems with AutoCAD previewer text, you should configure some URL to indicate an extra fonts repository used in AutoCAD files. Simply copy the files to some URL that can be resolved by desktop users and take into consideration that font file names should be in lower case (rename if you have uppercase font file names). You can also enable AutoCAD previewer debug mode.

```
dxfg.applet.debug.window=false
dxfg.applet.default.bigfont=arial
dxfg.applet.default.font=arial
dxfg.applet.fontdir=http://fonts.openkm.com
```

Since: OpenKM 6.0



Available only in OpenKM Professional.

Minimum search characters

You can set the minimum characters required to execute a search. If less characters are available the search button will not be enabled. By default the minimum search characters is set to 3.

```
min.search.characters = 3
```

Since: OpenKM 6.0.0

Execution timeout

The maximum permitted time to execute a command like OCR, Antivirus, etc. After this time the child process is killed. Time given in minutes.

```
system.execution.timeout=5
```

Since: OpenKM 6.0.1

Restrict by file name

You can avoid uploading of documents by name or extension. For example, don't permit upload of AVI multimedia videos.

```
restrict.file.name=*.avi
```

You can also configure several filename rules:

```
restrict.file.name=*~;*.bak;*.avi
```

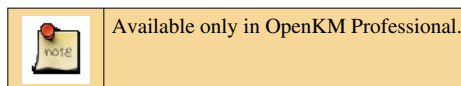
Since: OpenKM 6.2

Managed text extraction

In previous OpenKM releases, the text extraction was a transparent process where you can't do anything with the configuration. Since OpenKM 6.0 you have several configuration options to adapt the document text extraction to your needs. Let's see these parameters:

- **managed.text.extraction.batch**: Integer. How many documents are processed every time the text extractor worker is awoken.
- **managed.text.extraction.pool.size**: Integer. How many threads are launched concurrently every time. This number should be smaller or equal to the number of CPUs in the server.
- **managed.text.extraction.pool.threads**: Integer. How many threads will be executed in every time. If *managed.text.extraction.pool.size* is 8 and *managed.text.extraction.pool.threads* is set to 16 then all these 16 threads will be launched using this pool. If every task take the same time to complete, in theory every pool thread will execute two tasks. Of course this is not true in the real world.
- **managed.text.extraction.pool.timeout**: Minutes. How many minutes should the worker wait for the text extraction pool to complete the tasks. After these minutes, the remaining pool threads are interrupted.
- **managed.text.extraction.schedule**: Minutes. How often the text extractor worker is awoken.
- **managed.text.extraction.concurrent**: Boolean. If the concurrent text extraction should be enabled. By default the text extraction is serial, but you can improve the extraction performance if you have several CPUs.

Since: OpenKM 6.0.1



Document version numeration

Since OpenKM 6.2 we support different numeration schemas. By default OpenKM uses **com.openkm.vernum.MajorMinorVersionNumerationAdapter** which is similar to Jackrabbit implementation. But also have some other already implemented:

Numeration Adapter	First Version	Next Versions
PlainVersionNumerationAdapter	1	2, 3, 4, etc.
MajorMinorVersionNumerationAdapter	1.0	1.1, 1.2, 1.3, etc.
BranchVersionNumerationAdapter	1	2, 3, 2.1, 2.2, etc.

You can change the default **Numeration Adapter** using the **version.numeration.adapter** configuration property. You need to restart OpenKM every time you change this property. Also note that if already there are documents in the repository, changing the Numeration Adapter may cause problems because they are not compatible between them.

Some Numeration Adapters like **PlainVersionNumerationAdapter** may have additional configuration parameters like **version.numeration.format** which is used to define the numeration number format. By default this is "%d" but you can change to "%02d" to create versions like "01", "02", etc.

If you change the version numeration adapter you also need to update the existing numerations:

```
update NodeDocumentVersion ndv set ndv.name = '01' where ndv.name='1.0';
update NodeDocumentVersion ndv set ndv.name = '02' where ndv.name='1.1';
update NodeDocumentVersion ndv set ndv.name = '03' where ndv.name='1.2';
...
```

Since: OpenKM 6.2

Session keep alive

Some administration tasks take a long time, like repository checker and rebuild Lucene indexes. This implies that the user session may be lost. For this reason these kinds of operations are forced to maintain the session alive. You can configure the keep alive interval using the configuration property **keep.session.alive.interval**. By default is set to 5 minutes.

Since: OpenKM Professional 6.2.9

Activity log actions

The table where activity log is stored may grow quickly. OpenKM can log a lot of information related to the activity of the users, but sometimes these actions don't need to be logged and fill your activity log table. For this reason there is a configuration property called **activity.log.actions** where you can define which actions to log. By default this is set to the most common or interesting actions. You can use regular expressions to define these actions. Read Java Regex Tutorial ^[2] for more info about Java regular expressions.

Since: OpenKM Professional 6.2.10

Safe trash purge

When you delete a document or folder from OpenKM, it goes to user trash. If the user empties the trash, the documents are permanently lost. But now you can configure a place where the documents will go after being purged from trash. You can configure a directory in the server filesystem where an export of the trash be placed, using the **repository.purgatory.home** configuration property.

Since: OpenKM Professional 6.2.10

Tomcat Connector encoding

Default Tomcat Connector encoding is set to "ISO-8859-1" but you can change it with the property:

```
URIEncoding="UTF-8"
```

OpenKM should have the same encoding for proper input. So, you have the **tomcat.connector.uri.encoding** configuration property. By default this is set to "ISO-8859-1", but if you set the "URIEncoding" parameter in Tomcat to "UTF-8", you also need to set this OpenKM configuration property to "UTF-8".

Since: OpenKM Professional 6.2.12

References

[1] <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

[2] <http://www.vogella.com/articles/JavaRegularExpressions/article.html>

Article Sources and Contributors

Administration Guide *Source:* <http://wiki.openkm.com/index.php?oldid=10348> *Contributors:* Anonymous, Jllort, Pavila

Administration access *Source:* <http://wiki.openkm.com/index.php?oldid=5634> *Contributors:* Anonymous, Jllort, Pavila

Configuration view *Source:* <http://wiki.openkm.com/index.php?oldid=5635> *Contributors:* Anonymous, Jllort, Pavila

Mime types *Source:* <http://wiki.openkm.com/index.php?oldid=8171> *Contributors:* Anonymous, Jllort, Pavila

Statistics *Source:* <http://wiki.openkm.com/index.php?oldid=5637> *Contributors:* Anonymous, Jllort, Pavila

Scripting - OpenKM 6.2 *Source:* <http://wiki.openkm.com/index.php?oldid=9095> *Contributors:* Pavila

Property Groups administration *Source:* <http://wiki.openkm.com/index.php?oldid=8414> *Contributors:* Anonymous, Jllort, Nigel.pvg, Pavila

Property Groups *Source:* <http://wiki.openkm.com/index.php?oldid=8416> *Contributors:* Anonymous, Jllort, Pavila

Logged users *Source:* <http://wiki.openkm.com/index.php?oldid=2790> *Contributors:* Jllort, Pavila

Message queue *Source:* <http://wiki.openkm.com/index.php?oldid=6122> *Contributors:* Jllort

User administration *Source:* <http://wiki.openkm.com/index.php?oldid=8336> *Contributors:* Anonymous, Jllort, Pavila

Profiles *Source:* <http://wiki.openkm.com/index.php?oldid=8170> *Contributors:* Anonymous, Jllort, Pavila

Database query *Source:* <http://wiki.openkm.com/index.php?oldid=5647> *Contributors:* Anonymous, Jllort, Pavila

Reports *Source:* <http://wiki.openkm.com/index.php?oldid=9942> *Contributors:* Anonymous, Jllort, Pavila

Activity log *Source:* <http://wiki.openkm.com/index.php?oldid=5649> *Contributors:* Anonymous, Jllort, Pavila

Workflow *Source:* <http://wiki.openkm.com/index.php?oldid=8862> *Contributors:* Anonymous, Jllort, Pavila

Automation *Source:* <http://wiki.openkm.com/index.php?oldid=10726> *Contributors:* Jllort, Pavila

Enable automation *Source:* <http://wiki.openkm.com/index.php?oldid=10728> *Contributors:* Jllort, Pavila

Extend automation *Source:* <http://wiki.openkm.com/index.php?oldid=10246> *Contributors:* Jllort

Extend automation 6.4 *Source:* <http://wiki.openkm.com/index.php?oldid=10245> *Contributors:* Jllort

Extend automation 6.0 *Source:* <http://wiki.openkm.com/index.php?oldid=10233> *Contributors:* Jllort

Crontab *Source:* <http://wiki.openkm.com/index.php?oldid=9449> *Contributors:* Anonymous, Jllort, Pavila

OMR templates *Source:* <http://wiki.openkm.com/index.php?oldid=9634> *Contributors:* Jllort, Pavila

OCR templates *Source:* <http://wiki.openkm.com/index.php?oldid=10725> *Contributors:* Jllort, Pavila

Extend OCR field parsers *Source:* <http://wiki.openkm.com/index.php?oldid=10260> *Contributors:* Jllort

Thesaurus *Source:* <http://wiki.openkm.com/index.php?oldid=8423> *Contributors:* Jllort, Megamansgo.ok, Pavila

Folder style admin *Source:* <http://wiki.openkm.com/index.php?oldid=10337> *Contributors:* Jllort

Generate thesaurus *Source:* <http://wiki.openkm.com/index.php?oldid=5653> *Contributors:* Anonymous, Jllort, Pavila

Language *Source:* <http://wiki.openkm.com/index.php?oldid=8169> *Contributors:* Anonymous, Jllort, Pavila

Repository export *Source:* <http://wiki.openkm.com/index.php?oldid=5657> *Contributors:* Anonymous, Jllort, Pavila

Utilities admin *Source:* <http://wiki.openkm.com/index.php?oldid=10360> *Contributors:* Jllort

Utilites admin check mail *Source:* <http://wiki.openkm.com/index.php?oldid=10357> *Contributors:* Jllort

Application configuration *Source:* <http://wiki.openkm.com/index.php?oldid=10665> *Contributors:* Anonymous, Jllort, Pavila, Sean.fmq, Tanyona.be

Image Sources, Licenses and Contributors

[illegible]

[illegible]

File:Okm_admin_064.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_064.png *License:* unknown *Contributors:* Jllort

File:Okm_admin_067.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_067.png *License:* unknown *Contributors:* Jllort

File:Translate.png *Source:* <http://wiki.openkm.com/index.php?title=File:Translate.png> *License:* unknown *Contributors:* Jllort

File:Okm_admin_065.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_065.png *License:* unknown *Contributors:* Jllort

File:Export.png *Source:* <http://wiki.openkm.com/index.php?title=File:Export.png> *License:* unknown *Contributors:* Jllort

File:Okm_admin_027.jpeg *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_027.jpeg *License:* unknown *Contributors:* Jllort

File:Okm_admin_097.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_097.png *License:* unknown *Contributors:* Jllort

File:Okm_admin_098.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_098.png *License:* unknown *Contributors:* Jllort

File:Okm_admin_099.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_099.png *License:* unknown *Contributors:* Jllort

File:Okm_admin_100.png *Source:* http://wiki.openkm.com/index.php?title=File:Okm_admin_100.png *License:* unknown *Contributors:* Jllort

File:Folder.png *Source:* <http://wiki.openkm.com/index.php?title=File:Folder.png> *License:* unknown *Contributors:* Jllort

File:Document.png *Source:* <http://wiki.openkm.com/index.php?title=File:Document.png> *License:* unknown *Contributors:* Jllort